User's Guide

Publication Number 33120-90006 (*order as 33120-90104 manual set*)
Edition 6, March 2002

© Copyright Agilent Technologies, Inc. 1994-2002

For Safety information, Warranties, and Regulatory information, see the pages following the Index.

Agilent 33120A
15 MHz Function / Arbitrary Waveform Generator

Note: *Unless otherwise indicated, this manual applies to all Serial Numbers.*

The Agilent Technologies 33120A is a high-performance 15 MHz synthesized function generator with built-in arbitrary waveform capability. Its combination of bench-top and system features makes this function generator a versatile solution for your testing requirements now and in the future.

Convenient bench-top features

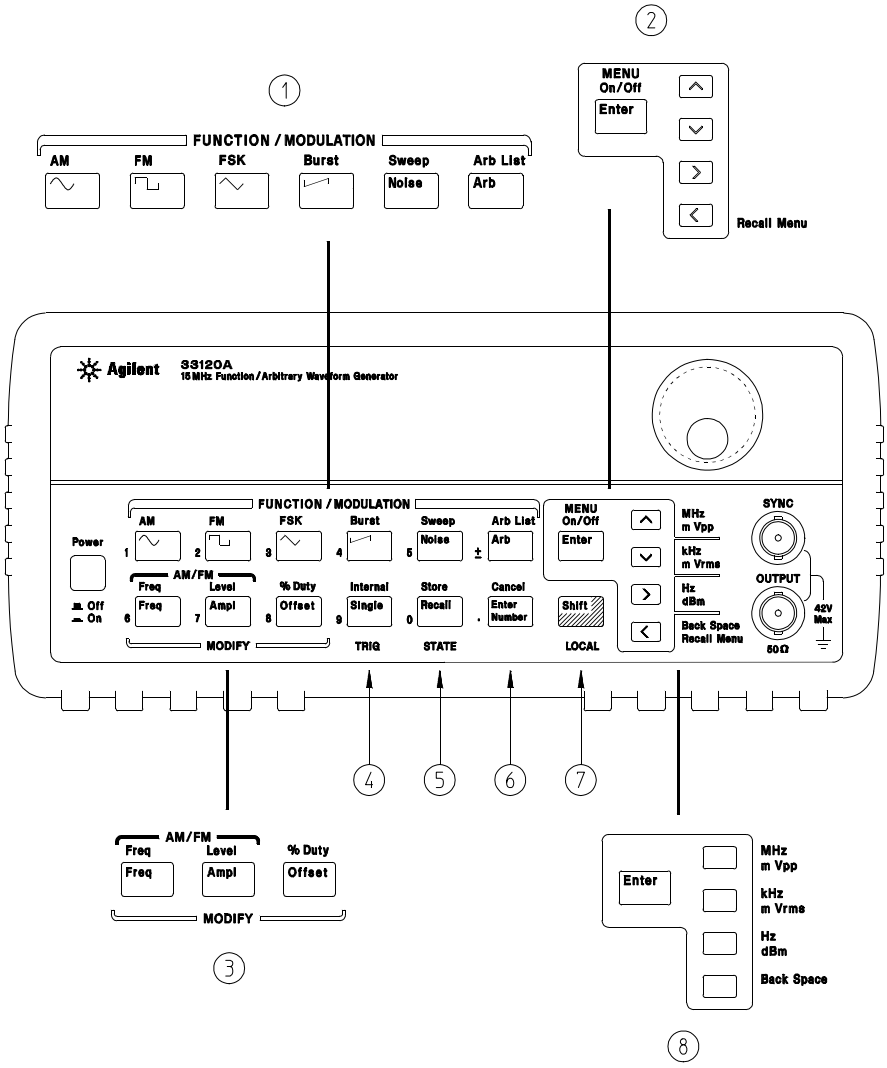
- 10 standard waveforms
- Built-in 12-bit 40 MSa/s arbitrary waveform capability
- Easy-to-use knob input
- Highly visible vacuum-fluorescent display
- Instrument state storage
- Portable, ruggedized case with non-skid feet

Flexible system features

- Four downloadable 16,000-point arbitrary waveform memories
- GPIB (IEEE-488) interface and RS-232 interface are standard
- SCPI (Standard Commands for Programmable Instruments) compatibility
- Agilent *IntuiLink Arb Waveform Generation Software* for Microsoft® Windows® included

Agilent 33120A
15 MHz Function / Arbitrary Waveform Generator

The Front Panel at a Glance

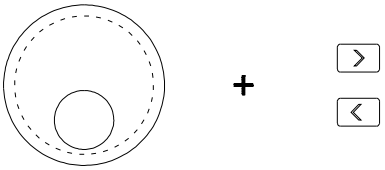


1 Function / Modulation keys
2 Menu operation keys
3 Waveform modify keys
4 Single / Internal Trigger key (*Burst and Sweep only)*
5 Recall / Store instrument state key
6 Enter Number key
7 Shift / Local key
8 Enter Number “units” keys

Front-Panel Number Entry

You can enter numbers from the front-panel using one of three methods.

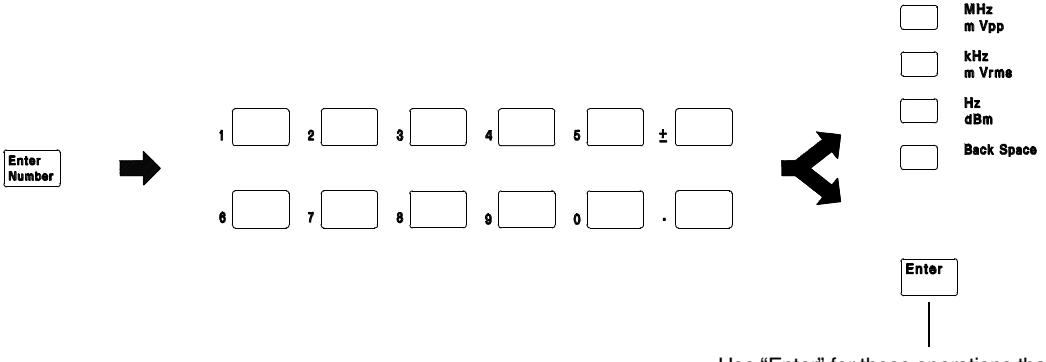
Use the knob and the arrow keys to modify the displayed number.



Use the arrow keys to edit individual digits.

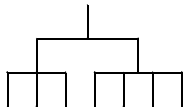
- **Increments** the flashing digit.
- **Decrements** the flashing digit.
- Moves the flashing digit **to the right**.
- Moves the flashing digit **to the left**.

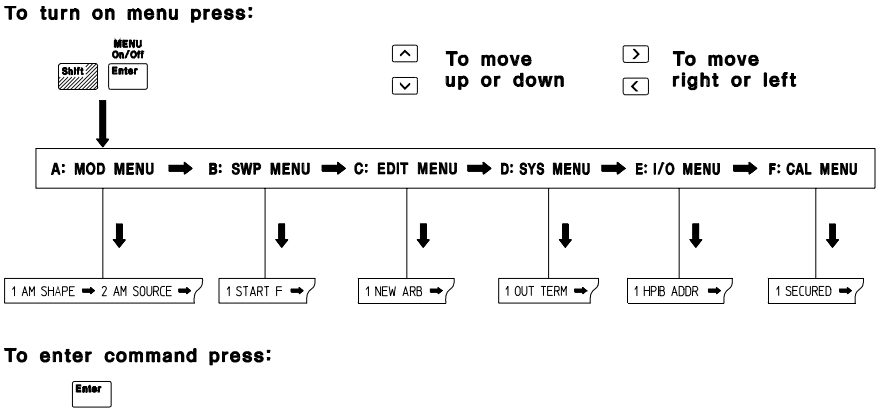
Use the "Enter Number" mode to enter a number with the appropriate units.



Use "Enter" for those operations that do not require units to be specified (AM Level, Offset, % Duty, and Store/Recall State).

The menu is organized in a top-down tree structure with three levels.





A: MODulation MENU

1: AM SHAPE ⇨ 2: AM SOURCE ⇨ 3: FM SHAPE ⇨ 4: BURST CNT ⇨ 5: BURST RATE ⇨

⇨ 6: BURST PHAS ⇨ 7: BURST SRC ⇨ 8: FSK FREQ ⇨ 9: FSK RATE ⇨ 10: FSK SRC

B: SWP (Sweep) MENU

1: START F ⇨ 2: STOP F ⇨ 3: SWP TIME ⇨ 4: SWP MODE

C: EDIT MENU*

1: NEW ARB ⇨ [2: POINTS] ⇨ [3: LINE EDIT] ⇨ [4: POINT EDIT] ⇨ [5: INVERT] ⇨ [6: SAVE AS] ⇨ 7: DELETE

* The commands enclosed in square brackets ([ ]) are “hidden” until you make a selection from the NEW ARB command to initiate a new edit session.

D: SYStem MENU

1: OUT TERM ⇨ 2: POWER ON ⇨ 3: ERROR ⇨ 4: TEST ⇨ 5: COMMA ⇨ 6: REVISION

E: Input / Output MENU

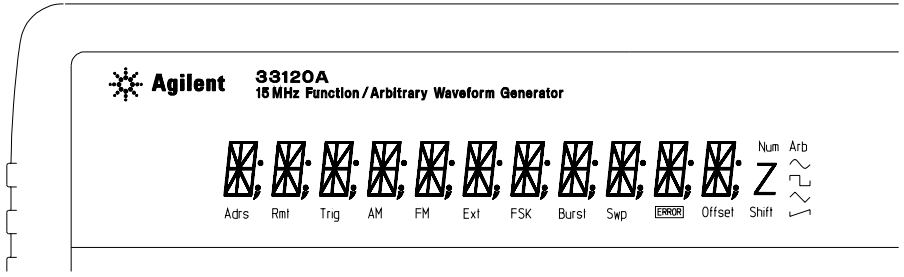
1: HPIB ADDR ⇨ 2: INTERFACE ⇨ 3: BAUD RATE ⇨ 4: PARITY ⇨ 5: LANGUAGE

F: CALibration MENU*

1: SECURED or [1: UNSECURED] ⇨ [2: CALIBRATE] ⇨ 3: CAL COUNT ⇨ 4: MESSAGE

* The commands enclosed in square brackets ([ ]) are “hidden” unless the function generator is UNSECURED for calibration.

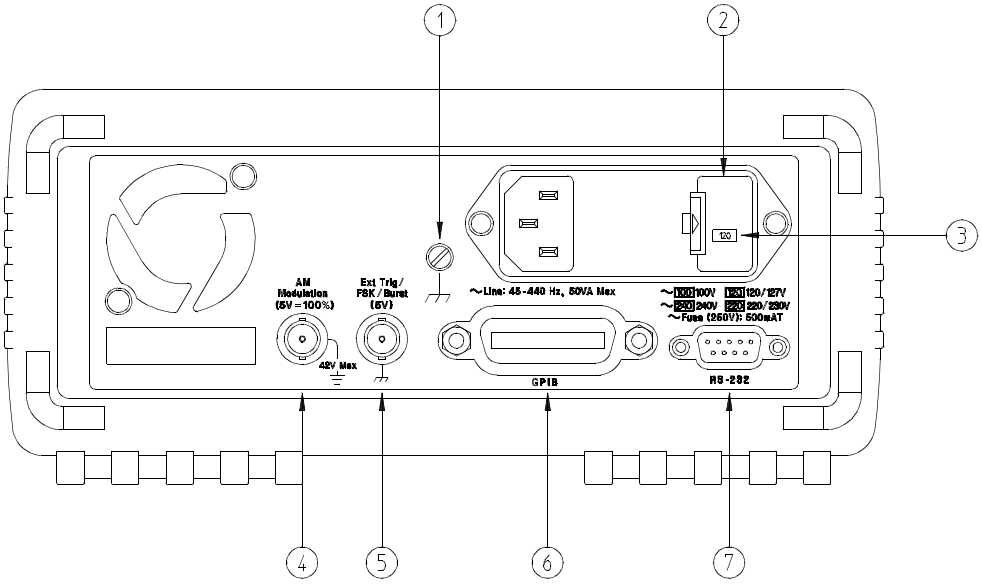
Display Annunciators



Adrs	Function generator is addressed to listen or talk over a remote interface.
Rmt	Function generator is in remote mode (remote interface).
Trig	Function generator is waiting for a single trigger or external trigger (Burst, Sweep).
AM	AM modulation is enabled.
FM	FM modulation is enabled.
Ext	Function generator is set for an external modulation source (AM, FSK, Burst).
FSK	FSK (frequency-shift keying) modulation is enabled.
Burst	Burst modulation is enabled.
Swp	Sweep mode is enabled.
ERROR	Hardware or remote interface command errors are detected.
Offset	The waveform is being output with an offset voltage.
Shift	"Shift" key has been pressed. *Press "Shift" again to turn off.*
Num	"Enter Number" mode is enabled. *Press "Shift-Cancel" to disable.*
Arb	Arbitrary waveform function is enabled.
∿	Sine waveform function is enabled.
⊓	Square waveform function is enabled.
△	Triangle waveform function is enabled.
◺	Ramp waveform function is enabled.

To review the display annunciators, hold down the Shift key as you turn on the function generator.

The Rear Panel at a Glance



1 Chassis ground
2 Power-line fuse-holder assembly
3 Power-line voltage setting
4 AM modulation input terminal
5 External Trigger / FSK / Burst modulation input terminal
6 GPIB (IEEE-488) interface connector
7 RS-232 interface connector

Use the front-panel Input / Output Menu to:

- Select the GPIB or RS-232 interface (see chapter 4).
- Set the GPIB bus address (see chapter 4).
- Set the RS-232 baud rate and parity (see chapter 4).

In This Book

Quick Start Chapter 1 prepares the function generator for use and helps you get familiar with a few of its front-panel features.

Front-Panel Menu Operation Chapter 2 introduces you to the front-panel menu and describes some of the function generator's menu features.

Features and Functions Chapter 3 gives a detailed description of the function generator's capabilities and operation. You will find this chapter useful whether you are operating the function generator from the front panel or over the remote interface.

Remote Interface Reference Chapter 4 contains reference information to help you program the function generator over the remote interface.

Error Messages Chapter 5 lists the error messages that may appear as you are working with the function generator. Each listing contains enough information to help you diagnose and solve the problem.

Application Programs Chapter 6 contains several remote interface application programs to help you develop programs for your application.

Tutorial Chapter 7 discusses the fundamentals of signal generation and modulation techniques.

Specifications Chapter 8 lists the function generator's specifications.

For information on using the Phase-Lock Option for the 33120A, refer to the User's and Service Guide included with the Option 001.

*If you have questions relating to the operation of the 33120A, call **1-800-452-4844** in the United States, or contact your nearest Agilent Technologies Sales Office.*

If you believe your 33120A has failed, refer to "Operating Checklist", "Types of Service Available", and "Repackaging for Shipment" at the beginning of chapter 6 in the 33120A Service Guide.

Contents

Chapter 1 Quick Start

To prepare the function generator for use 15
If the function generator does not turn on 16
To adjust the carrying handle 18
To set the output frequency 19
To set the output amplitude 20
To set a dc offset voltage 21
To set the duty cycle 22
To output a stored arbitrary waveform 23
To output a dc voltage 24
To store the instrument state 25
To rack mount the function generator 27

Chapter 2 Front-Panel Menu Operation

Front-panel menu reference 31
A front-panel menu tutorial 33
To select the output termination 40
To output a modulated waveform 41
To output an FSK waveform 44
To output a burst waveform 47
To output a frequency sweep 49
To trigger a burst or sweep 51
To turn off the comma separator 52

Chapter 3 Features and Functions

Output Configuration 55
Amplitude Modulation (AM) 71
Frequency Modulation (FM) 76
Burst Modulation 81
Frequency-Shift Keying (FSK) Modulation 90
Frequency Sweep 94
Triggering 98
Arbitrary Waveforms 103
System-Related Operations 109
Remote Interface Configuration 114
Calibration Overview 118
Power-On and Reset State 123

Chapter 4 Remote Interface Reference

SCPI Command Summary 127
Simplified Programming Overview 136
Using the APPLy Command 138
Output Configuration Commands 145
AM Modulation Commands 154
FM Modulation Commands 157
Burst Modulation Commands 160
Frequency-Shift Keying (FSK) Commands 167
Frequency Sweep Commands 170
Arbitrary Waveform Commands 174
Triggering 186
System-Related Commands 188
Calibration Commands 193
RS-232 Interface Configuration 195
RS-232 Interface Commands 200
The SCPI Status Registers 201
Status Reporting Commands 209
An Introduction to the SCPI Language 211
Halting an Output in Progress 216
To set the GPIB address 217
To select the remote interface 218
To set the baud rate 219
To set the parity 220
SCPI Conformance Information 221
IEEE-488 Conformance Information 225

Chapter 5 Error Messages

Execution Errors 229
Self-Test Errors 237
Calibration Errors 238
Arbitrary Waveform Errors 240

Chapter 6 Application Programs

BASIC Programs 244
C Language Programs 244
QuickBASIC Language Programs 247
Using the APPLy Command 248
Using the Low-Level Commands 252
Downloading an Arbitrary Waveform over GPIB 255
Using the Status Registers 261
Downloading an Arbitrary Waveform over RS-232 267

Chapter 7 Tutorial

Direct Digital Synthesis 273
Signal Imperfections 276
Creating Arbitrary Waveforms 278
Output Amplitude Control 280
Floating Signal Generators 282
Attributes of AC Signals 283
Modulation 287

Chapter 8 Specifications

Frequency Characteristics 298
Sinewave Spectral Purity 298
Signal Characteristics 298
Output Characteristics 298
Modulation Characteristics 299
Frequency Sweep 299
System Characteristics 299
General Specifications 300
Product Dimensions 301

Index 303

Declaration of Conformity 309

Quick Start

Quick Start

One of the first things you will want to do with your function generator is to become acquainted with its front panel. We have written the exercises in this chapter to prepare the function generator for use and help you get familiar with some of the front-panel operations.

The front panel has two rows of keys to select various functions and operations. Most keys have a *shifted* function printed in *blue* above the key. To perform a shifted function, press [Shift] (the **Shift** annunciator will turn on). Then, press the key that has the desired label above it. For example, to select the AM (amplitude modulation) function, press [Shift] [AM] (the shifted version of the [~] key).

If you accidentally press [Shift], just press it again to turn off the **Shift** annunciator.

Most keys also have a number printed in *green* next to the key. To enable the number mode, press [Enter Number] (the **Num** annunciator will turn on). Then, press the keys that have the desired numbers printed next to them. For example, to select the number "10", press [Enter Number] [1] [0] (next to the [~] and [Recall] keys).

If you accidentally press [Enter Number], just press [Shift] [Cancel] to turn off the **Num** annunciator.

To prepare the function generator for use

The following steps help you verify that the function generator is ready for use.

1 Check the list of supplied items.

Verify that you have received the following items with your function generator. If anything is missing, contact your nearest Agilent Technologies Sales Office.

- ☐ One power cord.
- ☐ One RS-232 serial cable.
- ☑ This *User's Guide*.
- ☐ One *Service Guide*.
- ☐ One folded *Quick Reference* card.
- ☐ Certificate of Calibration.
- ☐ Agilent IntuiLink Arb Waveform Generation Software.

2 Connect the power cord and turn on the function generator.

The front-panel display will light up while the function generator performs its power-on self-test. The GPIB bus address is displayed. Notice that the function generator powers up in the *sine wave* function at 1 kHz with an amplitude of 100 mV peak-to-peak (into a 50Ω termination).

To review the power-on display with all annunciators turned on, hold down Shift *as you turn on the function generator.*

3 Perform a *complete* self test.

The *complete* self-test performs a more extensive series of tests than those performed at power-on. Hold down Shift as you press the Power switch to turn on the function generator; *hold down the key for more than 5 seconds*. The self-test will begin when you release the key.

If the self-test is successful, "PASS" is displayed on the front panel. If the self-test is not successful, "FAIL" is displayed and the **ERROR** annunciator turns on. See the *Service Guide* for instructions on returning the function generator to Agilent for service.

If the function generator does not turn on

Use the following steps to help solve problems you might experience when turning on the function generator. If you need more help, see the *Service Guide* for instructions on returning the function generator to Agilent for service.

1 Verify that there is ac power to the function generator.

First, verify that the function generator's Power switch is in the "On" position. Also, make sure that the power cord is firmly plugged into to the power module on the rear panel. You should also make sure that the power source you plugged the function generator into is energized.

2 Verify the power-line voltage setting.

The line voltage is set to the proper value for your country when the function generator is shipped from the factory. Change the voltage setting if it is not correct. The settings are: 100, 120, 220, or 240 Vac (for 230 Vac operation, use the 220 Vac setting).

See the next page if you need to change the line-voltage setting.

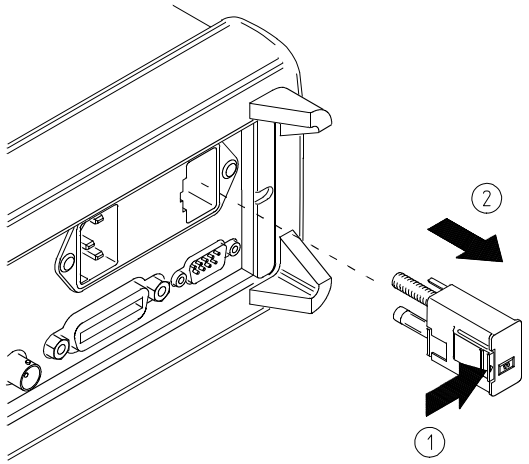
3 Verify that the power-line fuse is good.

The function generator is shipped from the factory with a 500 mAT fuse installed. This is the correct fuse for all line voltages.

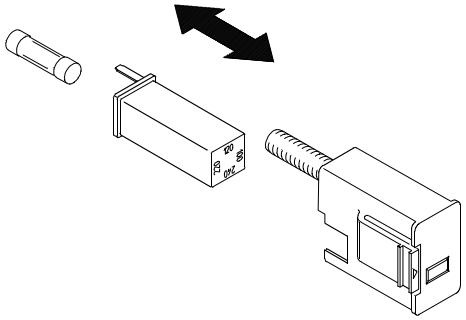
See the next page if you need to change the power-line fuse.

To replace the 500 mAT fuse, order Agilent part number 2110-0458.

1 ***Remove the power cord***. Remove the fuse-holder assembly from the rear panel.

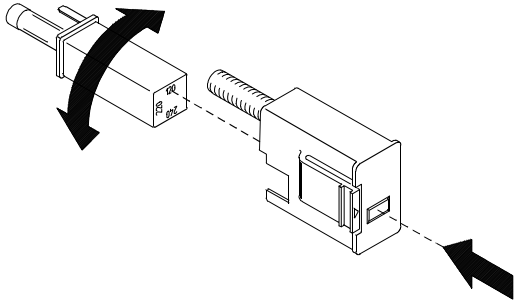


2 Remove the line-voltage selector from the assembly.



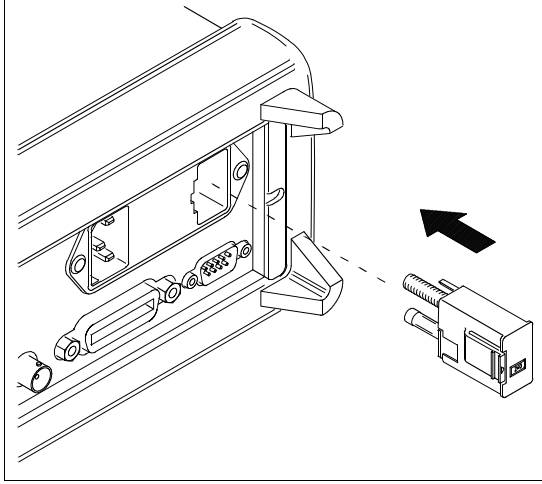
Fuse: 500 mAT (for all line voltages)
Part Number: 2110-0458

3 Rotate the line-voltage selector until the correct voltage appears in the window.



100, 120, 220 (230), or 240 Vac

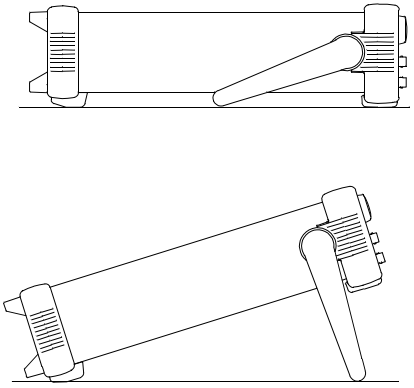
4 Replace the fuse-holder assembly in the rear panel.



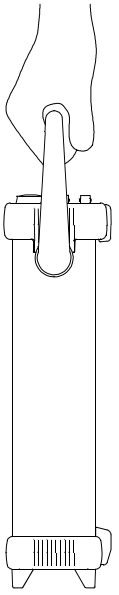
Verify that the correct line voltage is selected and the power-line fuse is good.

To adjust the carrying handle

To adjust the position, grasp the handle by the sides and *pull outward*. Then, rotate the handle to the desired position.



Bench-top viewing positions



Carrying position

To set the output frequency

At power-on, the function generator outputs a sine wave at 1 kHz with an amplitude of 100 mV peak-to-peak (into a 50Ω termination).
The following steps show you how to change the frequency to 1.2 MHz.

Freq

1 Enable the *frequency modify* mode.

The displayed frequency is either the power-on value or the previous frequency selected. When you change functions, the same frequency is used if the present value is valid for the new function.

```
1.000,000,0 KHz
```

Enter Number

1 . 2

2 Enter the magnitude of the desired frequency. [1]

Notice that the **Num** annunciator turns on and "ENTER NUM" flashes on the display, indicating that the number mode is enabled.

```
1.2
```

To cancel the number mode, press Shift Cancel .

^ MHz
m Vpp

3 Set the units to the desired value.

The units are selected using the arrow keys on the right side of the front panel. As soon as you select the units, the function generator outputs the waveform with the displayed frequency. *To turn off the flashing digit, move the cursor to the left of the display using the arrow keys.*

```
1.200,000,0 MHz
```

[1] *You can also use the knob and arrow keys to enter a number.*
See "Front-Panel Number Entry" on page 3 for more information.

To set the output amplitude

At power-on, the function generator outputs a sine wave with an amplitude of 100 mV peak-to-peak (into a 50Ω termination).
The following steps show you how to change the amplitude to 50 mVrms.

Ampl

1 Enable the *amplitude modify* mode.

The displayed amplitude is either the power-on value or the previous amplitude selected. When you change functions, the same amplitude is used if the present value is valid for the new function.

```
  100.0    mVPP
```

Enter Number

5 0

2 Enter the magnitude of the desired amplitude. [1]

Notice that the **Num** annunciator turns on and "ENTER NUM" flashes on the display, indicating that the number mode is enabled.

```
  50
```

To cancel the number mode, press Shift Cancel .

Shift

∨ kHz
m Vrms

3 Set the units to the desired value.

The units are selected using the arrow keys on the right side of the front panel. As soon as you select the units, the function generator outputs the waveform with the displayed amplitude. *To turn off the flashing digit, move the cursor to the left of the display using the arrow keys.*

```
  50.00    mVRMS
```

[1] *You can also use the knob and arrow keys to enter a number.*
See "Front-Panel Number Entry" on page 3 for more information.

To set a dc offset voltage

At power-on, the function generator outputs a sine wave with a dc offset voltage of 0 volts (into a 50Ω termination). *The following steps show you how to change the offset to –1.5 mVdc.*

Offset

1 Enable the *offset modify* mode.

The displayed offset voltage is either the power-on value or the previous offset selected. When you change functions, the same offset is used if the present value is valid for the new function.

```
+0.000    VDC
```

Enter Number

± 1 . 5

2 Enter the magnitude of the desired offset. ◎[1]

Notice that the **Num** annunciator turns on and "ENTER NUM" flashes on the display, indicating that the number mode is enabled. Notice that ± toggles the displayed value between **+** and **–** .

```
-1.5
```

To cancel the number mode, press Shift Cancel .

Shift

∨ kHz
m Vrms

3 Set the units to the desired value.

At this point, the function generator outputs the waveform with the displayed offset. Notice that the **Offset** annunciator turns on, indicating that the waveform is being output with an offset. The annunciator will turn on when the offset is any value other than 0 volts. *To turn off the flashing digit, move the cursor to the left of the display using the arrow keys.*

```
-01.50    mVDC
```

[1] *You can also use the knob and arrow keys to enter a number.*
See "Front-Panel Number Entry" on page 3 for more information.

To set the duty cycle

Applies only to square waves. At power-on, the duty cycle for square waves is 50%. You can adjust the duty cycle for a square waveform from 20% to 80%, in increments of 1% (for frequencies above 5 MHz, the range is 40% to 60%). *The following steps show you how to change the duty cycle to 45%.*

[⊓⊔]

1 Select the square wave function.

Notice that the ⊓⊔ annunciator turns on, indicating that the square wave function is enabled.

[Shift] [% Duty]

2 Enable the *duty cycle modify* mode.

The displayed duty cycle is either the power-on value or the previous value selected.

```
50 % DUTY
```

This message appears on the display for approximately 10 seconds. Repeat this step as needed.

[Enter Number]
[4] [5]

3 Enter the desired duty cycle. [1]

Notice that the **Num** annunciator turns on and "ENTER NUM" flashes on the display, indicating that the number mode is enabled.

```
45
```

To cancel the number mode, press [Shift] [Cancel] .

[Enter]

4 Output the waveform with the displayed duty cycle.

```
45 % DUTY
```

[1] *You can also use the knob and arrow keys to enter a number. See "Front-Panel Number Entry" on page 3 for more information.*

To output a stored arbitrary waveform

There are five built-in arbitrary waveforms stored in non-volatile memory for your use. You can output these waveforms directly from non-volatile memory. *The following steps show you how to output an "exponential rise" waveform from memory.*

Shift Arb List

1 Display the list of arbitrary waveforms.

The list contains the five built-in arbitrary waveforms (sinc, negative ramp, exponential rise, exponential fall, and cardiac). The list may also contain up to four user-defined arbitrary waveform names. The first choice on this level is "SINC".

This message appears on the display for approximately 10 seconds. Repeat this step as needed.

> >

2 Move across to the EXP_RISE choice. [1]

Enter

3 Select and output the displayed arbitrary waveform.

Notice that the **Arb** annunciator turns on, indicating that the output is an arbitrary waveform. The waveform is output using the present settings for frequency, amplitude, and offset unless you change them.

The selected waveform is now assigned to the Arb *key. Whenever you press this key, the selected arbitrary waveform is output.*

[1] *You can also use the knob to scroll left or right through the choices in the list. See "Front-Panel Number Entry" on page 3 for more information.*

To output a dc voltage

In addition to generating waveforms, you can also output a dc voltage in the range ± 5 Vdc (into a 50Ω termination). *The following steps show you how to output +155 mVdc.*

1 Press the Offset **key and *hold it down for more than 2 seconds*.**

To enter the dc voltage mode, press the Offset key or any key in the top row of function keys and ***hold it down for more than 2 seconds***. The displayed voltage is either the power-on value or the previous offset voltage selected.

```
      DCV
```

```
  +0.000    VDC
```

Enter Number

1 5 5

2 Enter the magnitude of the desired voltage. [1]

Notice that the **Num** annunciator turns on and "ENTER NUM" flashes on the display, indicating that the number mode is enabled.

```
  155
```

To cancel the number mode, press Shift Cancel .

Shift

∨ kHz
m Vrms

3 Set the units to the desired value.

At this point, the function generator outputs the displayed dc voltage. Notice that the **Offset** annunciator turns on (all other annunciators are off), indicating that a dc voltage is being output. The annunciator will turn on when the offset is any value other than 0 volts.

```
  +155.0    mVDC
```

[1] *You can also use the knob and arrow keys to enter a number. See "Front-Panel Number Entry" on page 3 for more information.*

To store the instrument state

You can store up to three different instrument states in non-volatile memory. This enables you to recall the entire instrument configuration with just a few key presses from the front panel. *The following steps show you how to store and recall a state.*

1 Set up the function generator to the desired configuration.

The state storage feature "remembers" the function, frequency, amplitude, dc offset, duty cycle, as well as any modulation parameters.

Shift Store

2 Turn on the state storage mode.

Three memory locations (numbered 1, 2, and 3) are available to store instrument configurations. The instrument configuration is stored in *non-volatile* memory and is remembered when power has been off.

```
STORE  1
```

This message appears on the display for approximately 10 seconds. Repeat this step as needed.

^

3 Store the instrument state in memory location "2". [1]

Use the up and down arrow keys to select the memory location.

```
STORE  2
```

To cancel the store operation, press Shift Store *again or let the display time-out after 10 seconds.*

Enter

4 Save the instrument state.

The instrument state is now stored. To recall the stored state, turn to the next page.

[1] *You can also use the knob or "enter number" mode to enter a memory location. See "Front-Panel Number Entry" on page 3 for more information.*

To verify that the state was stored properly, you can turn the power off before recalling the state.

Recall

5 Recall the stored instrument state.

To recall the stored state, you must use the same memory location used previously to store the state. Use the up and down arrow keys to change the displayed storage location.

To cancel the restore operation, press Recall *again.*

This message appears on the display for approximately 10 seconds. Repeat this step as needed.

Enter

6 Restore the instrument state.

The function generator should now be configured in the same state as when you stored the setup on the previous page.

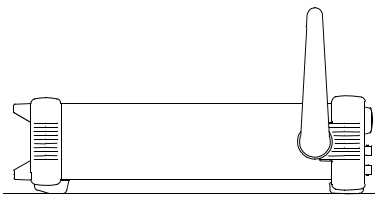
When power is turned off, the function generator automatically stores its state in memory location "0". You can recall the power-down state, but you cannot store the state to location "0" from the front panel.

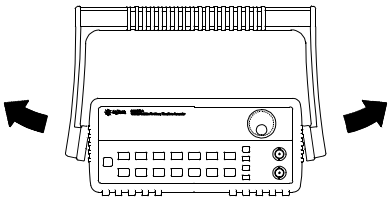
Use the POWER ON LAST STATE command in the SYS MENU to automatically recall the power-down state when power is turned on. See chapter 2 for more information on using the front-panel menus.

To rack mount the function generator

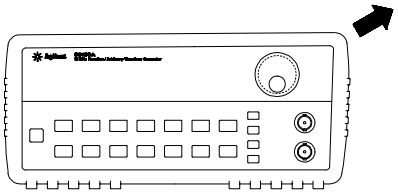
You can mount the function generator in a standard 19-inch rack cabinet using one of three optional kits available. Instructions and mounting hardware are included with each rack-mounting kit. Any Agilent *System II* instrument of the same size can be rack-mounted beside the 33120A Function Generator.

Remove the carrying handle, and the front and rear rubber bumpers, before rack-mounting the function generator.

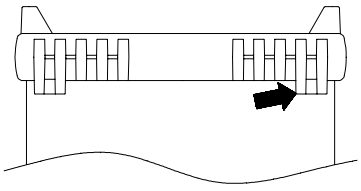




To remove the handle, rotate it to the vertical position and pull the ends outward.



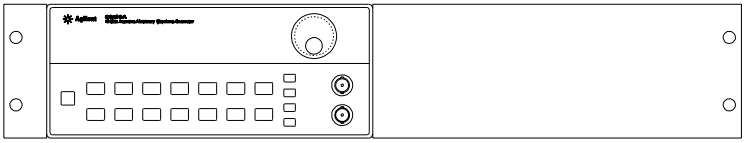
Front



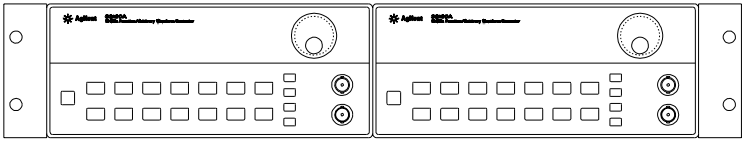
Rear (bottom view)

To remove the rubber bumper, stretch a corner and then slide it off.

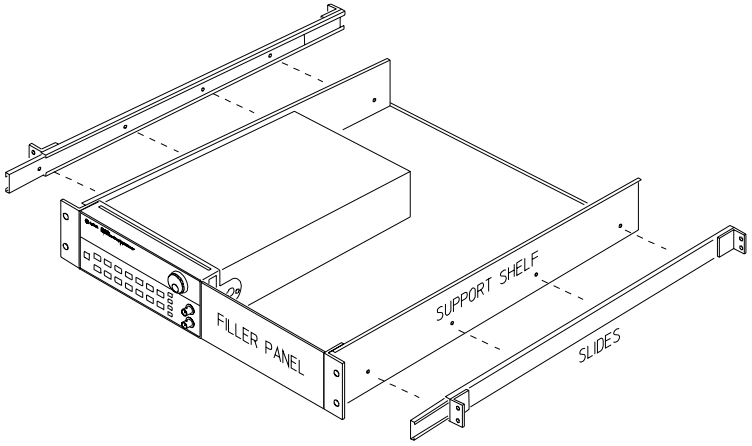
To rack mount the function generator



To rack mount a single instrument, order adapter kit 5063-9240.



To rack mount two instruments side-by-side, order lock-link kit 5061-9694 and flange kit 5063-9212.



To install one or two instruments in a sliding support shelf, order shelf 5063-9255, and slide kit 1494-0015 (for a single instrument, also order filler panel 5002-3999).

2

Front-Panel Menu Operation

Front-Panel Menu Operation

By now you should be familiar with some of the basic features of the front panel. Chapter 1 shows you how to prepare the function generator for use and describes a few of the front-panel features. If you are not familiar with this information, we recommend that you read chapter 1, "Quick Start," starting on page 13.

Chapter 2 introduces you to the use of the front-panel menu. This chapter *does not* give a detailed description of every front-panel key or menu operation. It does, however, give you a good overview of the front-panel menu and many front-panel operations. See chapter 3 "Features and Functions," starting on page 53, for a complete discussion of the function generator's capabilities and operation.

If you purchased the Phase-Lock Option for the 33120A, an additional menu (**G: PHASE MENU**) *is available from the front panel. For information on using the Phase-Lock Option, refer to the User's and Service Guide included with Option 001.*

Front-panel menu reference

A: MODulation MENU

1: AM SHAPE ⇨ 2: AM SOURCE ⇨ 3: FM SHAPE ⇨ 4: BURST CNT ⇨ 5: BURST RATE ⇨

⇨ 6: BURST PHAS ⇨ 7: BURST SRC ⇨ 8: FSK FREQ ⇨ 9: FSK RATE ⇨ 10: FSK SRC

1:	**AM SHAPE**	Selects the shape of the AM modulating waveform.
2:	**AM SOURCE**	Enables or disables the internal AM modulating source.
3:	**FM SHAPE**	Selects the shape of the FM modulating waveform.
4:	**BURST CNT**	Sets the number of cycles per burst (1 to 50,000 cycles).
5:	**BURST RATE**	Sets the burst rate in Hz for an internal burst source.
6:	**BURST PHAS**	Sets the starting phase angle of a burst (-360 to +360 degrees).
7:	**BURST SRC**	Selects an internal or external gate source for burst modulation.
8:	**FSK FREQ**	Sets the FSK “hop” frequency.
9:	**FSK RATE**	Selects the internal FSK rate between the carrier and FSK frequency.
10:	**FSK SRC**	Selects an internal or external source for the FSK rate.

B: SWP (Sweep) MENU

1: START F ⇨ 2: STOP F ⇨ 3: SWP TIME ⇨ 4: SWP MODE

1:	**START F**	Sets the start frequency in Hz for sweeping.
2:	**STOP F**	Sets the stop frequency in Hz for sweeping.
3:	**SWP TIME**	Sets the repetition rate in seconds for sweeping.
4:	**SWP MODE**	Selects linear or logarithmic sweeping.

C: EDIT MENU *

1: NEW ARB ⇨ [2: POINTS] ⇨ [3: LINE EDIT] ⇨ [4: POINT EDIT] ⇨ [5: INVERT] ⇨ [6: SAVE AS] ⇨ 7: DELETE

1:	**NEW ARB**	Initiates a new arb waveform or loads the selected arb waveform.
2:	**POINTS**	Sets the number of points in a new arb waveform (8 to 16,000 points).
3:	**LINE EDIT**	Performs a linear interpolation between two points in the arb waveform.
4:	**POINT EDIT**	Edits the individual points of the selected arb waveform.
5:	**INVERT**	Inverts the selected arb waveform by changing the sign of each point.
6:	**SAVE AS**	Saves the current arb waveform in non-volatile memory.
7:	**DELETE**	Deletes the selected arb waveform from non-volatile memory.

* The commands enclosed in square brackets ([ ]) are “hidden” until you make a selection from the NEW ARB command to initiate a new edit session.

D: SYStem MENU

1: OUT TERM ⇨ 2: POWER ON ⇨ 3: ERROR ⇨ 4: TEST ⇨ 5: COMMA ⇨ 6: REVISION

1: OUT TERM	Selects the output termination (50Ω or high impedance).
2: POWER ON	Enables or disables automatic recall of the power-down state.
3: ERROR	Retrieves errors from the error queue (up to 20 errors).
4: TEST	Performs a complete self-test.
5: COMMA	Enables or disables a comma separator between digits on the display.
6: REVISION	Displays the function generator's firmware revision codes.

E: Input / Output MENU

1: HPIB ADDR ⇨ 2: INTERFACE ⇨ 3: BAUD RATE ⇨ 4: PARITY ⇨ 5: LANGUAGE

1: HPIB ADDR	Sets the GPIB bus address (0 to 30).
2: INTERFACE	Selects the GPIB or RS-232 interface.
3: BAUD RATE	Selects the baud rate for RS-232 operation.
4: PARITY	Selects even, odd, or no parity for RS-232 operation.
5: LANGUAGE	Verifies the interface language: SCPI.

F: CALibration MENU *

1: SECURED ⇨ [1: UNSECURED] ⇨ [2: CALIBRATE] ⇨ 3: CAL COUNT ⇨ 4: MESSAGE

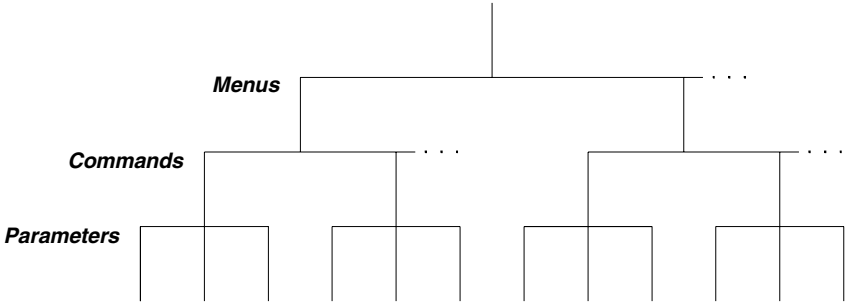
1: SECURED	The function generator is secured against calibration; enter code to unsecure.
1: UNSECURED	The function generator is unsecured for calibration; enter code to secure.
2: CALIBRATE	Performs individual calibrations; must be UNSECURED.
3: CAL COUNT	Reads the total number of times the function generator has been calibrated.
4: MESSAGE	Reads the calibration string (up to 11 characters) entered from remote.

* The commands enclosed in square brackets ([ ]) are "hidden" unless the function generator is UNSECURED for calibration.

A front-panel menu tutorial

This section is a step-by-step tutorial which shows you how to use the front-panel menu. We recommend that you spend a few minutes with this tutorial to get comfortable with the structure and operation of the menu.

The menu is organized in a top-down tree structure with three levels (*menus*, *commands*, and *parameters*). You move down [∨] or up [∧] the menu tree to get from one *level* to the next. Each of the three levels has several horizontal *choices* which you can view by moving left [<] or right [>] .



The menu is organized in a top-down tree structure with three levels.

- To *turn on* the menu, press [Shift] [Menu On/Off] .
- To *turn off* the menu, press [Shift] [Menu On/Off] .
- To *execute* a menu command, press [Enter] .
- To *recall* the last menu command that was executed, press [Shift] [Recall Menu] .
- To *turn off* the menu at any time without saving changes, press [Shift] [Cancel] .

Messages Displayed During Menu Use

TOP OF MENU You pressed [∧] while on the "MENUS" level; this is the top level of the menu and you cannot go any higher.

To turn off the menu, press [Shift] [Menu On/Off] *. To move across the choices on a level, press* [<] *or* [>] *. To move down a level, press* [∨] *.*

MENUS You are on the "MENUS" level. Press [<] or [>] to view the choices.

COMMANDS You are on the "COMMANDS" level. Press [<] or [>] to view the command choices within the selected menu group.

PARAMETER You are on the "PARAMETER" level. Press [<] or [>] to view and edit the parameter for the selected command.

MENU BOTTOM You pressed [∨] while on the "PARAMETER" level; this is the bottom level of the menu and you cannot go any lower.

To turn off the menu, press [Shift] [Menu On/Off] *. To move up a level, press* [∧] *.*

ENTERED The change made on the "PARAMETER" level is saved. This is displayed after you press [Enter] (Menu Enter) to execute the command.

MIN VALUE The value you specified on the "PARAMETER" level is too small for the selected command. The minimum value allowed is displayed for you to edit.

MAX VALUE The value you specified on the "PARAMETER" level is too large for the selected command. The maximum value allowed is displayed for you to edit.

EXITING You will see this message if you turn off the menu by pressing [Shift] [Menu On/Off] or [Shift] [Cancel] . You did not edit any values on the "PARAMETER" level and changes were NOT saved.

NOT ENTERED You will see this message if you turn off the menu by pressing [Shift] [Menu On/Off] or [Shift] [Cancel] . You did some editing of parameters but the changes were NOT saved. Press the [Enter] key (Menu Enter) to save changes made on the "PARAMETER" level.

Menu Example 1

The following steps show you how to turn on the menu, move up and down between levels, move across the choices on each level, and turn off the menu. *In this example, you will enable the function generator to automatically recall the power-down state when power is turned on.*

Shift

Menu On/Off

1 Turn on the menu.

You enter the menu on the "MENUS" level. The MOD MENU is your first choice on this level.

```
A: MOD MENU
```

> > >

2 Move across to the SYS MENU choice on this level. [1]

There are six menu group choices available on the "MENUS" level. Each choice has a letter prefix for easy identification (**A:** , **B:** , etc.).

```
D: SYS MENU
```

v

3 Move down to the "COMMANDS" level within the SYS MENU.

The OUT TERM command is your first choice on this level.

```
1: OUT TERM
```

>

4 Move across to the POWER ON command on this level. [1]

There are six command choices available in the SYS MENU. Each choice on this level has a number prefix for easy identification (**1:** , **2:** , etc.).

```
2: POWER ON
```

1 *You can also use the knob to scroll left or right through the choices on each level of the menu.*

v

5 Move down a level to the "PARAMETER" choices.

The first parameter choice is "DEFAULT" for the POWER ON command ("DEFAULT" is the factory setting and is stored in non-volatile memory).

```
DEFAULT
```

>

6 Move across to the "LAST STATE" choice. ◎[1]

There are two parameter choices for POWER ON.

```
LAST STATE
```

Enter

7 Save the change and turn off the menu.

The function generator beeps and displays a message to show that the change is now in effect. You are then exited from the menu.

```
ENTERED
```

8 Cycle the power to restore the default values.

Turn the function generator OFF and then ON. The default output state will now be in effect (1 kHz sine wave, 100 mV peak-to-peak, 50Ω termination).

[1] *You can also use the knob to scroll left or right through the choices on each level of the menu.*

Menu Example 2

The following example shows how to use the *recall menu* feature as a shortcut to set the POWER ON command back to its original setting. *You must perform the steps in Example 1 before you start this example.*

Shift

< Recall Menu

1 Use recall menu to return to the POWER ON command.

This returns you to the POWER ON command, which was the last command used before you exited the menu in the Example 1.

```
2: POWER ON
```

v

2 Move down to the "PARAMETER" choices.

The first parameter choice is "LAST STATE" (the current setting from Example 1).

```
LAST STATE
```

>

3 Move across to the "DEFAULT" choice. [1]

Set the parameter back to its original value.

```
DEFAULT
```

Enter

4 Save the change and turn off the menu.

The function generator beeps and displays a message to show that the change is now in effect. You are then exited from the menu.

```
ENTERED
```

[1] *You can also use the knob to scroll left or right through the choices on each level of the menu.*

Menu Example 3

Some commands in the menu require that you enter a numeric parameter value. The following steps show you how to enter a number in the menu. *For this example, you will set the number of cycles for the burst modulation mode.*

Shift

Menu On/Off

1 Turn on the menu.

You enter the menu on the "MENUS" level. The MOD MENU is your first choice on this level.

```
A: MOD MENU
```

∨

2 Move down to the "COMMANDS" level within the MOD MENU.

The AM SHAPE command is your first choice on this level.

```
1: AM SHAPE
```

> > >

3 Move across to the BURST CNT command on this level. [1]

There are ten command choices available in the MOD MENU.

```
4: BURST CNT
```

[1] *You can also use the knob to scroll left or right through the choices on each level of the menu.*

[∨]

4 Move down a level to edit the BURST CNT parameter.

The number of cycles should be "1" when you come to this point in the menu for the first time. For this example, you will set the number of cycles to "4". Notice that the **Burst** annunciator flashes, indicating that the displayed value is for the burst mode.

```
∧00001  CYC
```

When you see the flashing "∧" on the left side of the display, you can press [∧] *to abort the edit and return to the "COMMANDS" level.*

[<]

5 Move the flashing cursor over to edit the last digit.

Notice that the rightmost digit is flashing (the digit wraps around).

```
0000 1  CYC
```

[∧] [∧] [∧]

6 Increment the last digit until "4" is displayed. ◎[1]

You decrement or increment each digit independently.

```
0000 4  CYC
```

[Enter]

7 Save the change and turn off the menu.

The function generator beeps and displays a message to show that the change is now in effect. You are then exited from the menu.

```
ENTERED
```

[1] *You can also use the knob or "enter number" mode to enter a number. See "Front-Panel Number Entry" on page 3 for more information.*

To select the output termination

The function generator has a fixed output impedance of 50 ohms on the *OUTPUT* terminal. You can specify whether **you** are terminating the output into a 50Ω load or an open circuit. Incorrect impedance matching between the source and load will result in an output amplitude or dc offset which does not match the specified value.

Shift
Menu On/Off

1 Turn on the menu.

```
A: MOD MENU
```

> > >

2 Move across to the SYS MENU choice on this level. [1]

```
D: SYS MENU
```

∨

3 Move down a level to the OUT TERM command.

```
1: OUT TERM
```

∨ >

4 Move down a level and then across to the HIGH Z choice. [1]

With the output termination set to "HIGH Z", the function generator allows you to set the unloaded (open circuit) output voltage.

```
HIGH Z
```

Enter

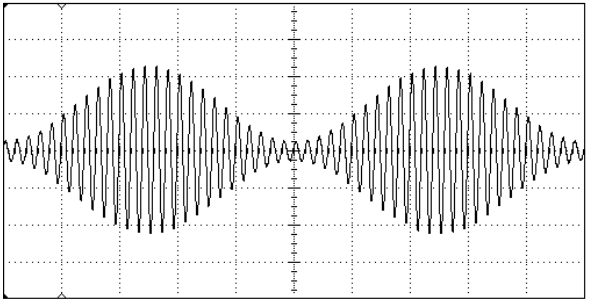
5 Save the change and turn off the menu.

The function generator beeps and displays a message to show that the change is now in effect. You are then exited from the menu.

[1] *You can also use the knob to scroll left or right through the choices on each level of the menu.*

To output a modulated waveform

A modulated waveform consists of a *carrier* and a *modulating* waveform. In AM (amplitude modulation), the amplitude of the carrier is varied by the amplitude of the modulating waveform. *For this example, you will output an AM waveform with 80% modulation depth. The carrier will be a 5 kHz sine wave and the modulating waveform will be a 200 Hz sine wave.*



[~]

1 Select the function, frequency, and amplitude of the carrier.

For the carrier waveform, you can select a sine, square, triangle, ramp, or arbitrary waveform. *For this example, select a **5 kHz** sine wave with an amplitude of **5 Vpp**.*

[Shift] [AM]

2 Select AM.

Notice that the **AM** annunciator turns on.

[Shift]

[<] Recall Menu

3 Use the menu to select the shape of the modulating waveform.

After you enable the AM function, the "recall menu" key will automatically take you to the AM SHAPE command in the MOD MENU.

```
1: AM SHAPE
```

[v]

4 Move down a level and verify that "SINE" is selected.

For the modulating waveform, you can select a sine, square, triangle, ramp, noise, or arbitrary waveform. For this example, you will modulate the carrier with a sine waveform. Notice that the **AM** annunciator flashes, indicating that the displayed parameter is for AM.

```
SINE
```

[Enter]

5 Save the change and turn off the menu.

The modulating waveform is now a sine waveform.

```
ENTERED
```

[Shift] [Freq]

6 Set the modulating frequency to 200 Hz.

Notice that the **AM** annunciator flashes, indicating that the displayed frequency is the modulating frequency for AM. Also notice that the modulating frequency is displayed with fewer digits than the carrier frequency. *For more information on editing numbers in the menu, refer to "Menu Example 3" earlier in this chapter.*

```
MOD  200.0  Hz
```

This message appears on the display for approximately 10 seconds. Repeat this step as needed.

Shift Level

7 Set the modulation depth to 80%.

Notice that the **AM** annunciator flashes, indicating that the displayed percentage is the AM depth (also called *percent modulation*).

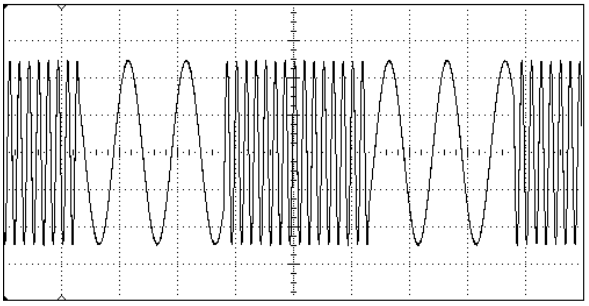
```
  080  %  DEPTH
```

This message appears on the display for approximately 10 seconds. Repeat this step as needed.

At this point, the function generator outputs the AM waveform with the specified modulation parameters.

To output an FSK waveform

You can configure the function generator to "shift" its output frequency between two preset values using FSK (frequency-shift keying) modulation. The rate at which the output shifts between the two frequencies (called the "carrier frequency" and the "hop frequency") is determined by the internal rate generator or the signal level on the rear-panel *FSK* terminal. *For this example, you will set the "carrier" frequency to 3 kHz and the "hop" frequency to 500 Hz, with an FSK rate of 100 Hz.*



[~]

1 Select the function, frequency, and amplitude of the carrier.

For the carrier waveform, you can select a sine, square, triangle, ramp, or arbitrary waveform. *For this example, select a **3 kHz** sine wave with an amplitude of **5 Vpp**.*

[Shift] [FSK]

2 Enable FSK.

Notice that the **FSK** annunciator turns on.

[Shift]

[<] Recall Menu

3 Use the menu to set the "hop" frequency.

After you enable the FSK mode, the "recall menu" key will automatically take you to the FSK FREQ command in the MOD MENU.

```
8: FSK FREQ
```

∨

4 Move down a level and set the "hop" frequency to 500 Hz.

Notice that the **FSK** annunciator flashes, indicating that the displayed parameter is for the FSK mode. Also notice that the hop frequency is displayed with fewer digits than the carrier frequency. *For more information on editing numbers in the menu, refer to "Menu Example 3" earlier in this chapter.*

```
∧500.0   Hz
```

Enter

5 Save the change and turn off the menu.

At this point, the carrier and hop frequencies are set and output.

Now, we will go back into the menu to set the FSK "shift" rate — this is the rate at which the function generator shifts between the carrier frequency and hop frequency.

Shift

< Recall Menu

6 Use the menu to set the FSK "shift" rate.

The "recall menu" key returns you to the FSK FREQ command, which was the last command used before you exited the menu.

```
8: FSK FREQ
```

[>]

7 Move across to the FSK RATE command.

```
9: FSK RATE
```

[v]

8 Move down a level and set the FSK "shift" rate to 100 Hz.

Notice that the **FSK** annunciator flashes, indicating that the displayed parameter is for the FSK mode. *For more information on editing numbers in the menu, refer to "Menu Example 3" earlier in this chapter.*

```
^100.0   Hz
```

[Enter]

9 Save the change and turn off the menu.

The function generator beeps and displays a message to show that the change is now in effect. You are then exited from the menu.

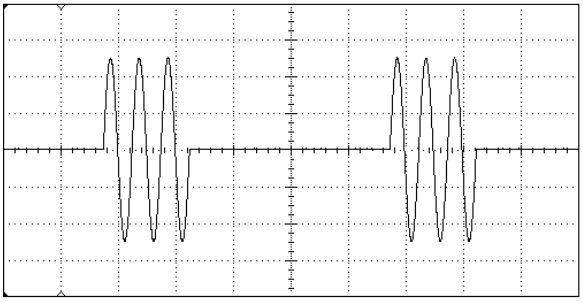
```
ENTERED
```

At this point, the function generator outputs the FSK waveform.

You can also use an external signal to shift between the carrier frequency and hop frequency. For more information, see "Frequency-Shift Keying (FSK) Modulation" in chapter 3.

To output a burst waveform

You can configure the function generator to output a waveform with a specified number of cycles, called a *burst*. You can output the burst at a rate determined by the internal rate generator or the signal level on the rear-panel *Ext Trig* terminal. *For this example, you will output a three-cycle sine wave. You will not change the other parameters from their default settings: internal burst source, 0 degree starting phase, and 100 Hz burst rate.*



[~] **1 Select the function, frequency, and amplitude for the burst.**

For bursts, you can select sine, square, triangle, ramp, or arbitrary waveforms. *For this example, select a **1 kHz** sine wave with an output amplitude of **5 Vpp**.*

[Shift] [Burst] **2 Enable the burst mode.**

Notice that the **Burst** annunciator turns on.

[Shift]

[<] Recall Menu **3 Use the menu to set the burst count.**

After you enable the burst mode, the "recall menu" key will automatically take you to the BURST CNT command in the MOD MENU.

```
4: BURST CNT
```

∨

4 Move down to the parameter level and set the count to "3".

Notice that the **Burst** annunciator flashes, indicating that the displayed parameter is for the burst mode. *For more information on editing numbers in the menu, refer to "Menu Example 3" earlier in this chapter.*

```
∧0000 3 CYC
```

You can also select an infinite burst count. Press the right or left arrow keys until the "CYC" units are flashing. Then, press the down arrow key to display "INFINITE".

Enter

5 Save the change and turn off the menu.

The function generator beeps and displays a message to show that the change is now in effect. You are then exited from the menu.

```
ENTERED
```

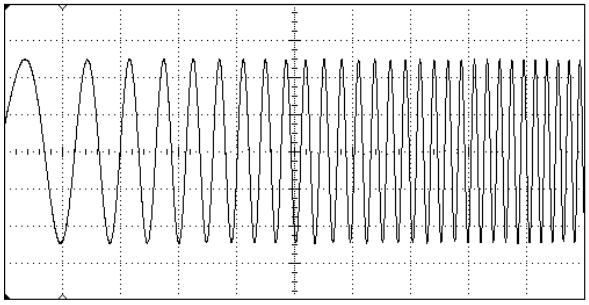
At this point, the function generator outputs a continuous three-cycle burst.

You can generate a single burst (with the specified burst count) by pressing the "Single" key from the front panel or applying a trigger pulse to the rear-panel Ext Trig terminal. For more information, see "To trigger a burst or sweep" on page 51.

You can also use an external gate signal to turn the output signal on and off. For more information, see "Burst Modulation" in chapter 3.

To output a frequency sweep

In the frequency sweep mode, the function generator "steps" from the *start frequency* to the *stop frequency* at a *sweep rate* which you specify. You can sweep up or down in frequency, and with either linear or logarithmic spacing. *For this example, you will output a swept sine wave from 50 Hz to 5 kHz. You will not change the other parameters from their default settings: internal sweep trigger, linear spacing, and 1 second sweep time.*



[~] **1 Select the function and amplitude for the sweep.**

For sweeps, you can select sine, square, triangle, ramp, or arbitrary waveforms. *For this example, select a sine wave with a **5 Vpp** amplitude.*

[Shift] [Sweep] **2 Enable the sweep mode.**

Notice that the **Swp** annunciator turns on.

[Shift] **3 Use the menu to select the start frequency.**

[<] Recall Menu

After you enable the sweep mode, the "recall menu" key will automatically take you to the START F command in the SWP MENU.

```
1: START F
```

∨

4 Move down a level and set the start frequency to 50 Hz.

Notice that the **Swp** annunciator flashes, indicating that the displayed parameter is for the sweep mode. *For more information on editing numbers in the menu, refer to "Menu Example 3" earlier in this chapter.*

```
∧50.00   Hz
```

Enter

5 Save the change and turn off the menu.

The start frequency is now set to 50 Hz.

Shift

< Recall Menu

6 Use recall menu to return to the START F command.

The "recall menu" key returns you to the START F command, which was the last command used before you exited the menu.

```
1: START F
```

>

7 Move across to the STOP F command.

```
2: STOP F
```

∨

8 Move down a level and set the stop frequency to 5 kHz.

```
∧5.000   KHz
```

Enter

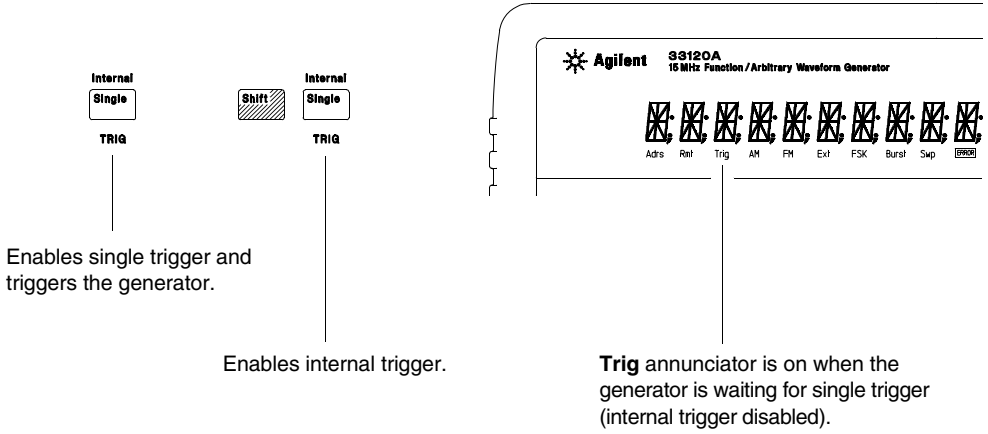
9 Save the change and turn off the menu.

At this point, the function generator outputs a continuous sweep from 50 Hz to 5 kHz.

You can generate one frequency sweep by pressing the "Single" key or applying a trigger pulse to the rear-panel Ext Trig terminal. For more information, see "To trigger a burst or sweep" on the next page.

To trigger a burst or sweep

You can issue triggers from the front-panel for burst modulation and frequency sweeps using *single* trigger or *internal* trigger.



- *Internal* or "automatic" triggering is enabled when you turn on the function generator. In this mode, the function generator outputs continuously when burst modulation or sweep is selected.
- *Single* triggering outputs one burst or initiates one frequency sweep each time you press [Single] . Continue pressing this key to re-trigger the function generator.
- Pressing [Single] to enable the *single* trigger mode also enables the *external* trigger mode. External triggering is like the single trigger mode except that you apply a trigger signal to the rear-panel *Ext Trig* terminal. The function generator is triggered on the rising edge of a TTL pulse.
- The [Single] key is disabled when in remote (the **Rmt** annunciator turns on when in remote) and when a function other than burst modulation or sweep is selected.

To turn off the comma separator

The function generator can display values on the front panel with or without a comma separator. The following steps show how to turn off the comma separator.

With comma separator
(factory setting)

Without comma separator

Shift
Menu On/Off

1 Turn on the menu.

> > >

2 Move across to the SYS MENU choice on the "MENUS" level.

v < <

3 Move down a level and then across to the COMMA command.

v >

4 Move down a level and then across to the "OFF" choice.

Enter

5 Save the change and turn off the menu.

The comma separator setting is stored in *non-volatile* memory, and *does not* change when power has been off or after a remote interface reset.

Features and Functions

Features and Functions

You will find that this chapter makes it easy to look up all the details about a particular feature of the function generator. Whether you are operating the function generator from the front panel or over the remote interface, this chapter will be useful. This chapter is divided into the following sections:

- Output Configuration, *starting on page 55*
- Amplitude Modulation (AM), *starting on page 71*
- Frequency Modulation (FM), *starting on page 76*
- Burst Modulation, *starting on page 81*
- Frequency-Shift Keying (FSK) Modulation, *starting on page 90*
- Frequency Sweep, *starting on page 94*
- Triggering, *starting on page 98*
- Arbitrary Waveforms, *starting on page 103*
- System-Related Operations, *starting on page 109*
- Remote Interface Configuration, *starting on page 114*
- Calibration Overview, *starting on page 118*
- Power-On and Reset State, *on page 123*

Some knowledge of the front-panel menu will be helpful before you read this chapter. If you have not already read chapter 2, "Front-Panel Menu Operation," starting on page 29, you may want to read it now. Chapter 4, "Remote Interface Reference," starting on page 125, lists the syntax for the SCPI commands available to program the function generator.

Throughout this manual, the following conventions are used for SCPI command syntax for remote interface programming.

- Square brackets (**[ ]**) indicate optional keywords or parameters.
- Braces (**{ }**) enclose parameters within a command string.
- Triangle brackets (**< >**) indicate that you must substitute a value for the enclosed parameter.
- A vertical bar (**|**) separates multiple parameter choices.

Output Configuration

This section contains information to help you configure the function generator for outputting waveforms. You may never have to change some of the parameters discussed here, but they are provided to give you the flexibility you might need.

Output Function

The function generator can output five standard waveforms including sine, square, triangle, ramp, and noise. You can also select one of five predefined arbitrary waveforms or download your own custom waveforms. You can internally modulate any of the standard waveforms (including arbitrary) using AM, FM, FSK, or burst modulation. Linear or logarithmic frequency sweeping is available for any of the standard waveforms (except noise) and arbitrary waveforms. *The default function is sine wave.*

- *Possible Conflict with Output Frequency*: The output frequency is automatically adjusted if you select a function whose maximum frequency is *less than* that of the currently active function. For example, if you output a 1 MHz sine wave and then change the function to triangle wave, the function generator will adjust the output to 100 kHz (the upper limit for triangle waves).

  From the front panel, "FREQ LIMIT" is displayed and the frequency is adjusted. From the remote interface, a *-221, "Settings conflict"* error is generated and the frequency is adjusted.

- *Possible Conflict with Output Amplitude*: The output amplitude is automatically adjusted if you select a function whose maximum amplitude is *less than* that of the currently active function. This conflict may arise when the output units are *Vrms* or *dBm* due to the differences in crest factor for the output functions. For example, if you output a 5 Vrms square wave (into 50 ohms) and then change the function to sine wave, the function generator will adjust the output amplitude to 3.535 Vrms (the upper limit for sine waves in Vrms).

  From the front panel, "AMPL LIMIT" is displayed and the amplitude is adjusted. From the remote interface, a *-221, "Settings conflict"* error is generated and the amplitude is adjusted.

Output Function
(continued)

- The following matrix shows which output functions are allowed with each modulation mode. Each "X" indicates a valid combination. If you change to a function that is not allowed with the selected modulation, the modulation mode is turned off.

	Sine	Square	Triangle	Ramp	Noise	Arb
AM Carrier	X	X	X	X		X
AM Modulating Wave	X	X	X	X	X	X
FM Carrier	X	X	X	X		X
FM Modulating Wave	X	X	X	X	X	X
FSK Modulation	X	X	X	X		X
Burst Modulation	X	X	X	X		X
Frequency Sweep	X	X	X	X		X

- *Front-Panel Operation*: To select a function, press any key in the top row of function keys. Press [Arb] to output the arbitrary waveform currently selected (to scroll through the waveform choices and make a selection, press [Arb List]).

- To select ***dc volts*** from the front panel, press [Offset] and ***hold it down for more than 2 seconds***.

- *Remote Interface Operation*:

  ```
  FUNCtion:SHAPe {SINusoid|SQUare|TRIangle|RAMP|NOISe|USER|DC}
  ```

  You can also use the `APPLy` command to select the function, frequency, amplitude, and offset with a single command.

Output Frequency

As shown below, the output frequency range depends on the function currently selected. *The default frequency is 1 kHz for all functions.*

Function	Minimum Frequency	Maximum Frequency
Sine	100 μHz	15 MHz
Square	100 μHz	15 MHz
Triangle	100 μHz	100 kHz
Ramp	100 μHz	100 kHz
Built-In Arbs [1]	100 μHz	5 MHz

1 There are five built-in arbitrary waveforms stored in non-volatile memory: sinc, negative ramp, exponential rise, exponential fall, and cardiac.

For *arbitrary waveforms* that you create and download to memory, the maximum frequency depends on the number of points specified in the waveform. As shown below, the maximum output frequency decreases as you specify more points in the waveform. The five built-in arbitrary waveforms can be output at a maximum of 5 MHz.

Number of Arb Points	Minimum Frequency	Maximum Frequency
8 to 8,192 (8k)	100 μHz	5 MHz
8,193 to 12,287 (12k)	100 μHz	2.5 MHz
12,288 to 16,000	100 μHz	200 kHz

Output Frequency
(continued)

- *Possible Conflict with Function Change*: The output frequency is automatically adjusted if you select a function whose maximum frequency is *less than* that of the currently active function. For example, if you output a 1 MHz sine wave and then change the function to triangle wave, the function generator will adjust the output to 100 kHz (the upper limit for triangle waves).

  From the front panel, "FREQ LIMIT" is displayed and the frequency is adjusted. From the remote interface, a *-221, "Settings conflict"* error is generated and the frequency is adjusted.

- *Possible Conflict with Duty Cycle (square wave only)*: For output frequencies above 5 MHz, the duty cycle is limited to values between 40% and 60% (below 5 MHz, the range is 20% to 80%). The duty cycle is automatically adjusted if you select a frequency that is not valid with the present duty cycle. For example, if you set the duty cycle to 70% and then change the frequency to 8 MHz, the function generator will automatically adjust the duty cycle to 60% (the upper limit for this frequency).

  From the front panel, "% DUTY LIMIT" is displayed and the duty cycle is adjusted. From the remote interface, a *-221, "Settings conflict"* error is generated and the duty cycle is adjusted.

- *Front-Panel Operation*: To set the output frequency, press Freq . Then use the knob, arrow keys, or "Enter Number" mode to set the frequency.

- *Remote Interface Operation*:

  ```
  FREQuency {<frequency>|MINimum|MAXimum}
  ```

  You can also use the `APPLy` command to select the function, frequency, amplitude, and offset with a single command.

Output Amplitude

As shown below, the output amplitude range depends on the function currently selected and the output termination. *The default amplitude is 100 mVpp (into 50 ohms) for all functions.*

Function	Output Termination	Minimum Amplitude	Maximum Amplitude
Sine	50Ω	50 mVpp	10 Vpp
Square	50Ω	50 mVpp	10 Vpp
Triangle	50Ω	50 mVpp	10 Vpp
Ramp	50Ω	50 mVpp	10 Vpp
Noise	50Ω	50 mVpp	10 Vpp
Built-In Arbs [1]	50Ω	50 mVpp	10 Vpp
Sine	Open Circuit	100 mVpp	20 Vpp
Square	Open Circuit	100 mVpp	20 Vpp
Triangle	Open Circuit	100 mVpp	20 Vpp
Ramp	Open Circuit	100 mVpp	20 Vpp
Noise	Open Circuit	100 mVpp	20 Vpp
Built-In Arbs [1]	Open Circuit	100 mVpp	20 Vpp

[1] There are five built-in arbitrary waveforms stored in non-volatile memory: sinc, negative ramp, exponential rise, exponential fall, and cardiac.

- For *arbitrary waveforms*, the maximum amplitude will be limited if the data points do not span the full range of the output DAC (Digital-to-Analog Converter). For example, the built-in "SINC" waveform does not use the full range of values between ±1 and therefore its maximum amplitude is 6.084 Vpp (into 50 ohms).
- *Possible Conflict with Function Change*: The output amplitude is automatically adjusted if you select a function whose maximum amplitude is *less than* that of the currently active function. This conflict may arise when the output units are *Vrms* or *dBm* due to the differences in crest factor for the output functions. For example, if you output a 5 Vrms square wave (into 50 ohms) and then change the function to sine wave, the function generator will adjust the output amplitude to 3.535 Vrms (the upper limit for sine waves in Vrms).

  From the front panel, "AMPL LIMIT" is displayed and the amplitude is adjusted. From the remote interface, a *-221, "Settings conflict"* error is generated and the amplitude is adjusted.

Output Amplitude *(continued)*

- *Output Amplitude and Output Termination*: The output amplitude is automatically adjusted (and no error is generated) if you change the output termination. For example, if you set the amplitude to 10 Vpp and then change the termination from 50 ohms to "high impedance", the displayed amplitude will *double* to 20 Vpp. If you change from "high impedance" to 50 ohms, the displayed amplitude will drop in half. *See "Output Termination" on page 65 for more information.*
- *Offset Voltage Restrictions*: The output amplitude (in Vpp) and the dc offset voltage must obey the following restrictions. If the specified amplitude is not valid, the function generator will automatically adjust it to the maximum value allowed with the present offset voltage. (V*max* is either 10 volts for a high impedance termination or 5 volts for a 50 ohm termination; V*pp* is the output amplitude in volts peak-to-peak.)

$$|V_{offset}| + \frac{V_{pp}}{2} \leq V_{max} \quad and \quad |V_{offset}| \leq 2 \times V_{pp}$$

  From the front panel, "MAX VALUE" is displayed and the amplitude is adjusted. From the remote interface, a *-221, "Settings conflict"* error is generated and the amplitude is adjusted.
- A momentary glitch occurs in the output waveform at certain voltages due to output attenuator switching. This positive-going glitch occurs when the output voltage crosses the break-point voltage either from a lower voltage or a higher voltage. The voltages are shown below (for a 0 volt dc offset):

```
252 mVpp, 399 mVpp, 502 mVpp, 796 mVpp, 1 Vpp, 1.59 Vpp,
2.0 Vpp, 3.17 Vpp, 3.99 Vpp, 6.32 Vpp, 7.96 Vpp
```

- The output voltage will momentarily drop to 0 volts at certain voltages due to output relay switching. This occurs when the output voltage crosses the break-point voltage either from a lower voltage or a higher voltage. The voltages are shown below (for a 0 volt dc offset):

```
317 mVpp, 632 mVpp, 1.26 Vpp, 2.52 Vpp, 5.02 Vpp
```

- You can set the units for output amplitude to Vpp, Vrms, or dBm. *See "Output Units" on page 64 for more information.*

- For *dc volts*, the output level is actually controlled by setting the offset voltage. You can set the dc voltage to any value between ±5 Vdc into 50 ohms or ±10 Vdc into an open circuit. *See "DC Offset Voltage" on page 62 for more information.*

  *To select **dc volts** from the front panel, press* Offset and ***hold it down for more than 2 seconds***.

- *Front-Panel Operation*: To set the output amplitude, press Ampl . Then use the knob, arrow keys, or "Enter Number" mode to set the amplitude.

- *Remote Interface Operation*:

  ```
  VOLTage {<amplitude>|MINimum|MAXimum}
  ```

  You can also use the `APPLy` command to select the function, frequency, amplitude, and offset with a single command.

DC Offset Voltage

At power-on, the dc offset is set to 0 volts. You can set the offset to a positive or negative number with the restrictions shown below. If the specified offset voltage is not valid, the function generator will automatically adjust it to the maximum dc voltage allowed with the present amplitude. (V*max* is either 10 volts for a high impedance termination or 5 volts for a 50 ohm termination; V*pp* is the output amplitude in volts peak-to-peak.)

$$| V_{offset} | + \frac{V_{pp}}{2} \leq V_{max} \quad and \quad | V_{offset} | \leq 2 \times V_{pp}$$

From the front panel, "MAX VALUE" is displayed and the offset is adjusted. From the remote interface, a *-221, "Settings conflict"* error is generated and the offset is adjusted.

- *DC Offset and Output Termination*: The offset voltage is automatically adjusted (and no error is generated) if you change the output termination. For example, if you set the offset to 100 mVdc and then change the termination from 50 ohms to "high impedance", the displayed offset will *double* to 200 mVdc. If you change from "high impedance" to 50 ohms, the displayed offset will drop in half.
  See "Output Termination" on page 65 for more information.

- For *dc volts*, the output level is actually controlled by setting the offset voltage. You can set the dc voltage to any value between ±5 Vdc into 50 ohms or ±10 Vdc into an open circuit.

  To select ***dc volts*** *from the front panel, press* Offset and ***hold it down for more than 2 seconds***.

- For *arbitrary waveforms*, the **Offset** annunciator will turn on if the waveform data has an inherent offset present (if the average is not equal to zero). The function generator calculates the average of the data points and compares the average to zero volts. If the average is not within two DAC (Digital-to-Analog Converter) counts of zero volts, the **Offset** annunciator turns on.

- *Front-Panel Operation*: To set the dc offset, press Offset . Then use the knob, arrow keys, or "Enter Number" mode to set the offset.

- *Remote Interface Operation*:

  ```
  VOLTage:OFFSet {<offset>|MINimum|MAXimum}
  ```

  You can also use the `APPLy` command to select the function, frequency, amplitude, and offset with a single command.

Output Units

Applies only to output amplitude (does not affect offset). At power-on, the units for output amplitude are *volts peak-to-peak*.

- Output units: **Vpp**, Vrms, or dBm. *The default is Vpp*.
- The unit setting is stored in *volatile* memory; the units are set to "Vpp" when power has been off or after a remote interface reset.
- When switching from remote programming over the interface back to front-panel (local) operation, the output units are remembered. For example, if you select "Vrms" from the remote interface, the units are shown as "Vrms" on the front-panel display.
- *Front-Panel Operation*: Use the "Enter Number" units keys (the arrow keys located on the right side of the front panel) to select the output units. You can also convert from one unit to another from the front panel. For example, to convert 2 Vpp to its equivalent value in *Vrms*, press the following keys:

  Enter Number ∨ kHz / m Vrms

  The converted value is 707.1 mVrms.

- *Remote Interface Operation*:

```
VOLTage:UNIT {VPP|VRMS|DBM|DEFault}
```

Output Termination

Applies only to output amplitude and offset voltage. The function generator has a fixed output impedance of 50 ohms on the *OUTPUT* terminal. You can specify whether **you** are terminating the output into a 50 ohm load or an open circuit. Incorrect impedance matching between the function generator and your load will result in an amplitude or offset which does not match the specified signal level.

- Output termination: **50**Ω or High impedance. *The default is 50*Ω. See the table on page 59 for a list of amplitude limits for all functions.
- The output termination setting is stored in *volatile* memory; 50Ω is selected when power has been off or after a remote interface reset.
- The amplitude (or dc offset) is automatically adjusted (and no error is generated) if you change the output termination. For example, if you set the amplitude to 10 Vpp and then change the termination from 50 ohms to "high impedance", the displayed amplitude will *double* to 20 Vpp. If you change from "high impedance" to 50 ohms, the displayed amplitude will drop in half.
- If you specify a 50 ohm termination but are actually terminating into an open circuit, the displayed output will be *twice* the value specified. For example, if you set the offset to 100 mVdc (and specify a 50 ohm termination) but are actually terminating the output into an open circuit, the actual displayed offset will be 200 mVdc.
- *Front-Panel Operation*: Choose "50 OHM" or "HIGH Z" from the front-panel menu.

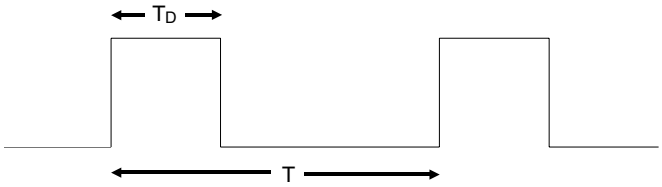
  ```
  1: OUT TERM  (SYS MENU)
  ```

- *Remote Interface Operation*:

  ```
  OUTPut:LOAD {50|INFinity|MINimum|MAXimum}
  ```

Duty Cycle

Applies only to square waves. Duty cycle is specified as a percentage and represents the amount of time per cycle that the square wave is high.



$$\text{Duty Cycle} = \frac{T_D}{T} \qquad (\text{where } T = \frac{1}{\text{frequency}})$$

- Duty cycle: 20% to 80%, in 1% increments (*frequency* ≤ 5 MHz).
  40% to 60%, in 1% increments (*frequency* > 5 MHz).
  The default is 50%.
- The duty cycle is stored in *volatile* memory; the duty cycle is set to 50% when power has been off or after a remote interface reset. The `APPLy` command automatically sets the duty cycle to 50% for square waves.
- Before attempting to set the duty cycle, you must enable the square wave function. From the front panel, "SQUARE ONLY" is displayed if square wave is not currently active. From the remote interface, no error is generated, but the specified duty cycle is remembered when you change to the square wave function.
- The duty cycle setting is remembered when you change from square wave to another function. When you return to the square wave function, the previous duty cycle is used.

- *Possible Conflict with Output Frequency*: The duty cycle is automatically adjusted if you select a frequency that is not valid with the present duty cycle. For example, if you set the duty cycle to 70% and then change the frequency to 8 MHz, the function generator will automatically adjust the duty cycle to 60% (the upper limit for this frequency).

  From the front panel, "% DUTY LIMIT" is displayed and the duty cycle is adjusted. From the remote interface, a *-221, "Settings conflict"* error is generated and the duty cycle is adjusted.

- *Front-Panel Operation*: To set the duty cycle, press Shift % Duty (the value appears on the display for approximately 10 seconds). Then use the knob, arrow keys, or "Enter Number" mode to set the duty cycle.

- *Remote Interface Operation*:

  ```
  PULSe:DCYCle {<percent>|MINimum|MAXimum}
  ```

  The `APPLy` command automatically sets the duty cycle to 50% for square waves.

SYNC Signal

A sync signal output is provided on the front-panel *SYNC* terminal. All of the standard output functions (except dc and noise) have an associated sync signal. For certain applications where you may not want to output the sync signal, you can disable the *SYNC* terminal.

The SYNC terminal can be enabled/disabled from the remote interface only.

- By default, the sync signal is routed to the *SYNC* terminal (enabled).
- When the sync signal is disabled, the output level on the *SYNC* terminal is indeterminate (it might be a TTL "high" or a TTL "low").
- For *sine*, *square*, *triangle*, and *ramp* waveforms, the sync signal is a TTL "high" when the waveform's output is positive, relative to zero volts (or the dc offset value). The signal is a TTL "low" when the output is negative, relative to zero volts (or the dc offset value).
- For *arbitrary* waveforms, a momentary TTL "high" pulse (> 200 ns) is output which corresponds to the first downloaded point in the waveform.
- For *AM* and *FM*, the sync signal is referenced to the modulating signal (not the carrier). A momentary TTL "high" pulse (> 200 ns) is output at each zero-crossing point of the modulating signal.
- For the *triggered burst mode*, a TTL "low" signal is output while the specified number of cycles is output (for the duration of the burst). After the specified number of cycles has been output, the sync signal goes "high" until the next burst.
- For the *external gated burst mode*, the sync signal is a TTL "high" when the output is positive, relative to zero volts (or the dc offset value). The signal is a TTL "low" when the output is negative, relative to zero volts (or the dc offset value).
- For *FSK*, a momentary TTL "high" pulse (> 200 ns) is output on the transition to the "hop" frequency.
- For *frequency sweeps*, the sync signal is a TTL "low" at the start of the sweep (when the start frequency is output) and is a TTL "high" at the end of the sweep (when the stop frequency is output).
- *Remote Interface Operation*:

```
OUTPut:SYNC {OFF|ON}
```
Setting is stored in volatile memory.

Instrument State Storage

You can store up to three different instrument states in non-volatile memory. This enables you to recall the entire instrument configuration with a single command from the remote interface or with just a few key presses from the front panel.

- Three memory locations (numbered 1, 2, and 3) are available to store instrument configurations. The state storage feature "remembers" the function (including arbitrary waveforms), frequency, amplitude, dc offset, duty cycle, as well as any modulation parameters. To recall a stored state, you must use the same memory location used previously to store the state.
- You cannot recall the instrument state from a memory location that was not previously specified as a storage location. For example, an error is generated if you attempt to recall from memory location "2" but have never stored to that location.

  From the front panel, "NOT STORED" is displayed if nothing is stored in the specified memory location. From the remote interface, a *+810, "State has not been stored"* error is generated if nothing is stored in the specified memory location.
- Any arbitrary waveforms downloaded to "VOLATILE" memory *are not* remembered. However, if an arbitrary waveform is being output from *non-volatile* memory when the state is stored, the waveform data is stored. The stored waveform is output when the instrument state is recalled.
- If you delete an arbitrary waveform after storing the state, the waveform data is lost and the function generator will not output the waveform when the state is recalled. The "SINC" waveform is output in place of the deleted waveform.

State Storage
(continued)

- When power is turned off, the function generator automatically stores its state in memory location "0". You can configure the function generator to automatically recall the power-down state when power is restored. The recall mode is *disabled* when the function generator is shipped from the factory.

  Select the POWER ON LAST STATE command from the SYS MENU to enable the power-down recall mode. Select POWER ON DEFAULT to disable the power-down recall mode. *See "Power-Down Recall Mode" on page 109 for more information.*

- From the remote interface, you can use memory location "0" to store a fourth instrument configuration (you cannot store to this memory location from the front panel). However, keep in mind that memory location "0" is automatically overwritten when the power is turned off.

- *Front-Panel Operation*:

  To store the state, press Shift Store . Then use the knob, arrow keys, or "Enter Number" mode to select the memory location.

  To restore the state, press Recall . Then use the knob, arrow keys, or "Enter Number" mode to select the memory location.

- *Remote Interface Operation*:

  ```
  *SAV {0|1|2|3}        State 0 is the instrument state at power down.
  *RCL {0|1|2|3}        States 1, 2, and 3 are user-defined states.
  ```

- From the remote interface, you can delete individual stored states and clear the memory location. If nothing is stored in the specified memory location, a *+810, "State has not been stored"* error is generated.

  ```
  MEMory:STATe:DELete {0|1|2|3}
  ```

Amplitude Modulation (AM)

A modulated waveform consists of a *carrier waveform* and a *modulating waveform*. In AM, the amplitude of the carrier is varied by the amplitude of the modulating waveform. The function generator will accept an internal modulating signal, an external modulating signal, or both.

For more information on the fundamentals of Amplitude Modulation, refer to chapter 7, "Tutorial".

To Select AM Modulation

- The **AM** annunciator turns on when AM is enabled.
- Only one modulation mode can be enabled at a time. When you enable AM, the previous modulation mode is turned off.
- *Front-Panel Operation*: Enable AM *before* setting up the other modulation parameters. Press Shift AM to output an AM waveform using the present settings for the carrier frequency, modulating frequency, output amplitude, and offset voltage.
- *Remote Interface Operation*: To ensure proper operation, you should enable AM *after* you have set up the other modulation parameters.

```
AM:STATe {OFF|ON}
```

Carrier Waveform Shape

- AM carrier shape: **Sine**, Square, Triangle, Ramp, or Arbitrary waveform. *The default is Sine.*
- You *cannot* use the noise function or dc volts as the carrier waveform.
- *Front-Panel Operation*: Press any of the front-panel function keys except [Noise] . For arbitrary waveforms, the [Arb] key selects the waveform that is currently downloaded.
- *Remote Interface Operation*:

```
FUNCtion:SHAPe {SINusoid|SQUare|TRIangle|RAMP|USER}
```

  You can also use the `APPLy` command to select the function, frequency, amplitude, and offset with a single command.

Carrier Frequency

- Carrier frequency: 100 μHz to 15 MHz (100 kHz for triangle and ramp). *The default is 1 kHz.*
- For *arbitrary waveforms*, the maximum carrier frequency depends on the number of points specified in the waveform. The five built-in arbitrary waveforms can be output at a maximum of 5 MHz.

Number of Arb Points	Minimum Frequency	Maximum Frequency
8 to 8,192 (8k)	100 μHz	5 MHz
8,193 to 12,287 (12k)	100 μHz	2.5 MHz
12,288 to 16,000	100 μHz	200 kHz

- *Front-Panel Operation*: Press the [Freq] key to select any valid frequency for the selected function (the present amplitude and offset voltage are used).
- *Remote Interface Operation*:

```
FREQuency {<frequency>|MINimum|MAXimum}
```

Modulating Waveform Shape

The function generator will accept an internal modulating signal, an external modulating signal, or both.

- Modulating waveform shape (internal source): **Sine**, Square, Triangle, Ramp, Noise, or Arbitrary waveform. *The default is Sine.*
- You can use the noise function as the modulating waveform. However, you *cannot* use the noise function or dc volts as the carrier waveform.
- *Front-Panel Operation*: After enabling AM, press *Recall Menu* to go directly to the AM SHAPE command in the menu.

```
1: AM SHAPE  (MOD MENU)
```

- *Remote Interface Operation*:

```
AM:INTernal:FUNCtion {SINusoid|SQUare|TRIangle|RAMP|NOISe|USER}
```

Modulating Waveform Frequency

The function generator will accept an internal modulating signal, an external modulating signal, or both.

- Modulating frequency (internal source): 10 mHz to 20 kHz. *The default is 100 Hz.*
- The sync signal for AM is referenced to the modulating signal (not the carrier). A momentary TTL "high" pulse (> 200 ns) is output at each zero-crossing point of the modulating signal. The signal is output from the front-panel *SYNC* terminal.
- *Front-Panel Operation*: Press [Shift] [Freq] to set the modulating frequency (the value is displayed for approximately 10 seconds).

  Before setting the modulating frequency from the front panel, you must have AM enabled. If you attempt to set the frequency when AM is not enabled, "SELECT AM, FM" is displayed.

- *Remote Interface Operation*:

```
AM:INTernal:FREQuency {<frequency>|MINimum|MAXimum}
```

Modulation Depth

The *modulation depth* is expressed as a percentage and represents the extent of the amplitude variation. At 0% modulation, the output amplitude is half of the selected value. At 100% modulation, the output amplitude equals the selected value.

- Modulation depth: 0% to 120%. *The default is 100%.*
- *Front-Panel Operation*: Press [Shift] [Level] to set the depth in percent (the value is displayed for approximately 10 seconds).

  Before setting the modulation depth from the front panel, you must have AM enabled. If you attempt to set the depth when AM is not enabled, "SELECT AM, FM" is displayed.

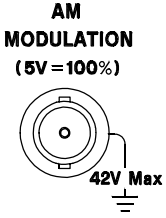
- *Remote Interface Operation*:

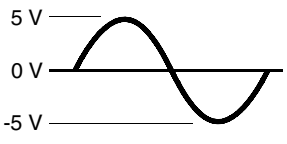
```
AM:DEPTh {<depth in percent>|MINimum|MAXimum}
```

Modulating Source

The function generator will accept an internal modulating signal, an external modulating signal, or both.

- Modulating source: Internal-External (both) or External only. *The default is Both (internal-external).*
- ***The External modulating source is always enabled.***
- When both sources are enabled (*internal-external*), the function generator adds the internal and external modulating signals (the carrier waveform is actually modulated with two waveforms).
- When the internal source is disabled (*external only*), the carrier waveform is modulated with the external waveform. The **Ext** annunciator turns on to indicate that the function generator is expecting a modulating signal on the rear-panel *AM Modulation* terminal.
- You apply the external modulating waveform to the *AM Modulation* terminal. The *modulation depth* is controlled by the signal level present (5 volts **peak** corresponds to 100% modulation).





- *Front-Panel Operation*:

```
2: AM SOURCE   (MOD MENU)
```

- *Remote Interface Operation*:

```
AM:SOURce {BOTH|EXTernal}
```

Frequency Modulation (FM)

A modulated waveform consists of a *carrier waveform* and a *modulating waveform*. In FM, the frequency of the carrier is varied by the frequency of the modulating waveform. The function generator will accept *only* an internal FM modulating signal (no external source is available).

For more information on the fundamentals of Frequency Modulation, refer to chapter 7, "Tutorial".

To Select FM Modulation

- The **FM** annunciator turns on when FM is enabled.
- Only one modulation mode can be enabled at a time. When you enable FM, the previous modulation mode is turned off.
- *Front-Panel Operation*: Enable FM *before* setting up the other modulation parameters. Press Shift FM to output an FM waveform using the present settings for the carrier frequency, modulating frequency, output amplitude, and offset voltage.
- *Remote Interface Operation*: To ensure proper operation, you should enable FM *after* you have set up the other modulation parameters.

```
FM:STATe {OFF|ON}
```

Carrier Waveform Shape

- FM carrier shape: **Sine**, Square, Triangle, Ramp, or Arbitrary waveform. *The default is Sine.*
- You *cannot* use the noise function or dc volts as the carrier waveform.
- *Front-Panel Operation*: Press any of the front-panel function keys except Noise . For arbitrary waveforms, the Arb key selects the waveform that is currently downloaded.
- *Remote Interface Operation*:

```
FUNCtion:SHAPe {SINusoid|SQUare|TRIangle|RAMP|USER}
```

  You can also use the `APPLy` command to select the function, frequency, amplitude, and offset with a single command.

Carrier Frequency

- Carrier frequency: 10 mHz to 15 MHz (100 kHz for triangle and ramp). *The default is 1 kHz.*
- For *arbitrary waveforms*, the maximum carrier frequency depends on the number of points specified in the waveform. The five built-in arbitrary waveforms can be output at a maximum of 5 MHz.

Number of Arb Points	Minimum Frequency	Maximum Frequency
8 to 8,192 (8k)	10 mHz	5 MHz
8,193 to 12,287 (12k)	10 mHz	2.5 MHz
12,288 to 16,000	10 mHz	200 kHz

- The carrier frequency must always be greater than or equal to the *peak frequency deviation*. If you attempt to set the carrier frequency to a value less than the deviation, the function generator will automatically adjust the carrier frequency to equal the present deviation.

  From the front panel, "MIN VALUE" is displayed and the carrier frequency is adjusted. From the remote interface, a *-221, "Settings conflict"* error is generated and the carrier frequency is adjusted.

- The sum of the carrier frequency and *peak frequency deviation* must be less than or equal to the maximum frequency for the selected function ***plus 100 kHz*** (15.1 MHz for sine and square, 200 kHz for triangle and ramp, and 5.1 MHz for arbitrary waveforms). If you attempt to set the carrier frequency to a value that is not valid, the function generator will automatically adjust the carrier frequency to equal the present deviation.

  From the front panel, "MAX VALUE" is displayed and the deviation is adjusted. From the remote interface, a *-221, "Settings conflict"* error is generated and the deviation is adjusted.

- *Front-Panel Operation*: Press the Freq key to select any valid frequency for the selected function (the present amplitude and offset voltage are used).

- *Remote Interface Operation*:

```
FREQuency {<frequency>|MINimum|MAXimum}
```

Modulating Waveform Shape

The function generator will accept *only* an internal modulating signal.

- Modulating waveform shape (internal source): **Sine**, Square, Triangle, Ramp, Noise, or Arbitrary waveform. *The default is Sine.*
- You can use the noise function as the modulating waveform. However, you *cannot* use the noise function or dc volts as the carrier waveform.
- *Front-Panel Operation*: After enabling FM, press *Recall Menu* to go directly to the FM SHAPE command in the menu.

```
3: FM SHAPE   (MOD MENU)
```

- *Remote Interface Operation*:

```
FM:INTernal:FUNCtion {SINusoid|SQUare|TRIangle|RAMP|NOISe|USER}
```

Modulating Waveform Frequency

The function generator will accept *only* an internal modulating signal.

- Modulating frequency (internal source): 10 mHz to 10 kHz. *The default is 10 Hz.*
- The sync signal for FM is referenced to the modulating signal (not the carrier). A momentary TTL "high" pulse (> 200 ns) is output at each zero-crossing point of the modulating signal. The signal is output from the front-panel *SYNC* terminal.
- *Front-Panel Operation*: Press Shift Freq to set the modulating frequency (the value is displayed for approximately 10 seconds).

  Before setting the modulating frequency from the front panel, you must have FM enabled. If you attempt to set the frequency when FM is not enabled, "SELECT AM, FM" is displayed.

- *Remote Interface Operation*:

```
FM:INTernal:FREQuency {<frequency>|MINimum|MAXimum}
```

Peak Frequency Deviation

The *peak frequency deviation* represents the variation in frequency of the modulating waveform from the carrier frequency.

- Peak frequency deviation: 10 mHz to 7.5 MHz. *The default is 100 Hz.*
- The *carrier frequency* must always be greater than or equal to the peak frequency deviation. If you attempt to set the deviation to a value greater than the carrier frequency (with FM enabled), the function generator will automatically adjust the deviation to equal the present carrier frequency.

  From the front panel, "MAX VALUE" is displayed and the deviation is adjusted. From the remote interface, a *-221, "Settings conflict"* error is generated and the deviation is adjusted.
- The sum of the *carrier frequency* and peak frequency deviation must be less than or equal to the maximum frequency for the selected function ***plus 100 kHz*** (15.1 MHz for sine and square, 200 kHz for triangle and ramp, and 5.1 MHz for arbitrary waveforms). If you attempt to set the deviation to a value that is not valid, the function generator will automatically adjust the deviation to the maximum value allowed with the present carrier frequency.

  From the front panel, "MAX VALUE" is displayed and the deviation is adjusted. From the remote interface, a *-221, "Settings conflict"* error is generated and the deviation is adjusted.

- *Front-Panel Operation*: Press Shift Level to set the peak deviation (the value is displayed for approximately 10 seconds).

  Before setting the deviation from the front panel, you must have FM enabled. If you attempt to set the deviation when FM is not enabled, "SELECT AM, FM" is displayed.

- *Remote Interface Operation*:

  `FM:DEViation {<`*peak deviation in Hz*`>|MINimum|MAXimum}`

Burst Modulation

You can configure the function generator to output a waveform with a specified number of cycles, called a *burst*. You can output the burst at a rate determined by the internal rate generator or an external signal applied to the rear-panel connector. *The function generator can produce a burst using sine, square, triangle, ramp, and arbitrary waveforms.*

For more information on the fundamentals of Burst Modulation, refer to chapter 7, "Tutorial".

You can use burst modulation in the following two modes. The function generator enables one burst mode at a time based on the *trigger source* and *burst source* that you select (see the table below).

- *Triggered Burst Mode*: In this mode, the function generator outputs a waveform with a specified number of cycles (*burst count*) each time a trigger is received. After the specified number of cycles has been output, the function generator waits for the next trigger. You can configure the function generator to accept an internal trigger or you can send the trigger externally by pressing the front-panel Single key or by applying a trigger signal to the rear-panel *Ext Trig* terminal.

  At power-on, the internally-triggered burst mode is enabled.

- *External Gated Burst Mode*: In this mode, the output waveform is either "on" or "off" based on the level of the external signal applied to the rear-panel *Ext Trig* terminal. When the gate signal is true, the function generator outputs a continuous waveform. When the gate signal is false, the output is turned off.

	Trigger Source (TRIG:SOUR)	**Burst Source** (BM:SOUR)	**Burst Count** (BM:NCYC)	**Burst Rate** (BM:INT:RATE)	**Burst Phase** (BM:PHAS)
External Triggered Mode	External	Internal	Available	Not Used	Available
Internal Triggered Mode	Internal	Internal	Available	Available	Available
External Gated Mode	Internal	External	Not Used	Not Used	Not Used

Burst Trigger Source

In the *triggered burst* mode, the function generator outputs a waveform with the specified number of cycles (*burst count*) each time a trigger is received. After the specified number of cycles has been output, the function generator waits for the next trigger while outputting no signal (zero volts or the dc offset level). *At power-on, the internally-triggered burst mode is enabled.*

- Trigger source: **Internal**, Single, or External. *The default is Internal.*
- In the triggered burst mode, the "burst source" is *Internal*.
- When the *internal* trigger source is selected, the frequency at which the burst is generated is determined by the *burst rate* parameter. The burst rate frequency defines the interval between bursts.
- Pressing Single to enable the *single* trigger mode also enables the *external* trigger mode. External triggering is like the front-panel single trigger mode except that you apply a trigger signal to the rear-panel *Ext Trig* terminal. The function generator is triggered on the *rising edge* of a TTL pulse.
- The **Trig** annunciator turns on when the function generator is waiting for a single or external trigger.
- When the *single* or *external* trigger source is selected, the *burst count* and *burst phase* remain in effect but the *burst rate* is ignored.
- When the *external gate* source is selected ("gated" burst mode), the specified trigger source is ignored. The external gated mode overrides the triggered mode source.
- *Front-Panel Operation*: Pressing Single to enable the single trigger mode also enables the external trigger mode. Press Shift Internal to enable the internal trigger mode.
- *Remote Interface Operation*:

```
TRIGger:SOURce {IMMediate|EXTernal|BUS}
```

See "Triggering" on page 98 for more information.

Burst Source

In the *external gated burst* mode, the output waveform is either "on" or "off" based on the level of the external signal applied to the rear-panel *Ext Trig* terminal. When the gate signal is true, the function generator outputs a continuous waveform. When the gate signal is false, the output is turned off.

- Burst source: **Internal** or External Gate. *The default is Internal.*
- When the *internal* burst source is selected, the external gated mode is disabled.
- When the *external gate* source is selected, the output is enabled or disabled based on the logic level of the gate signal applied to the *Ext Trig* terminal. When the gate signal is true (TTL high), the function generator outputs a continuous waveform. When the gate signal is false (TTL low), the output is turned off (zero volts or the dc offset level).
- When the *external gate* source is selected, the burst count, burst rate, burst phase, and burst trigger source are ignored (these parameters are used for the triggered burst mode only).
- The **Ext** annunciator turns on when the function generator is waiting for the external gate signal.

- *Front-Panel Operation*:

  7: BURST SRC **(MOD MENU)**

- *Remote Interface Operation*:

  ```
  BM:SOURce {INTernal|EXTernal}
  ```

To Select Burst Modulation

- The **Burst** annunciator turns on when burst modulation is enabled.
- Only one modulation mode can be enabled at a time. When you enable the burst mode, the previous modulation mode is turned off.
- *Front-Panel Operation*: Enable burst modulation *before* you set up the other modulation parameters. Press Shift Burst to output the burst using the present settings for the carrier frequency, output amplitude, and offset voltage.
- *Remote Interface Operation*: To ensure proper operation, you should enable the burst mode *after* you have set up the other modulation parameters.

```
BM:STATe {OFF|ON}
```

Burst Carrier Frequency

The *carrier frequency* defines the repetition rate of the burst waveform in the triggered and external gated modes. In the triggered mode, the number of cycles specified by the *burst count* are output at the frequency of the carrier signal. In the external gated mode, the carrier frequency is output when the external gate signal is true (TTL high).

Keep in mind that the carrier frequency is different than the "burst rate" which specifies the interval between bursts (triggered mode only).

- Carrier frequency: 10 mHz to 5 MHz (100 kHz for triangle and ramp). *The default is 1 kHz*. You can use sine, square, ramp, triangle, or arbitrary waveforms for the carrier waveshape.
- ***Be sure to note the restrictions for carrier frequency and burst count shown on the following pages.***

- For *arbitrary waveforms* used as the carrier waveform, the maximum frequency depends on the number of points specified in the waveform. The five built-in arbitrary waveforms can be output at a maximum of 5 MHz (*be sure to note the restrictions below).*

Number of Arb Points	Minimum Frequency	Maximum Frequency
8 to 8,192 (8k)	10 mHz	5 MHz
8,193 to 12,287 (12k)	10 mHz	2.5 MHz
12,288 to 16,000	10 mHz	200 kHz

- For *sine, square,* and *arbitrary* waveforms (does not apply to ramp and triangle waveforms), the relationship between the carrier frequency and the minimum burst count is shown below.

Carrier Frequency	Minimum Burst Count
10 mHz to 1 MHz	1
>1 MHz to 2 MHz	2
>2 MHz to 3 MHz	3
>3 MHz to 4 MHz	4
>4 MHz to 5 MHz	5

For sine, square, and arbitrary waveforms only.

- If you attempt to set the carrier frequency to a value that is not valid, the function generator will automatically adjust the frequency to the maximum value allowed with the present burst count.

  From the front panel, "MAX VALUE" is displayed and the carrier frequency is adjusted. From the remote interface, a *-221, "Settings conflict"* error is generated and the carrier frequency is adjusted.

Carrier Frequency *(continued)*

- For *all waveforms* used with burst, if the carrier frequency is set less than or equal to *100 Hz*, the following relationship applies.

$$\frac{\text{Burst Count}}{\text{Carrier Frequency}} \leq 500 \text{ seconds} \qquad \textit{For Carrier} \leq \textit{100 Hz}$$

  If you attempt to set the carrier frequency to a value that is not valid, the function generator will automatically adjust the frequency to the minimum value allowed with the present burst count.

  From the front panel, "MIN VALUE" is displayed and the frequency is adjusted. From the remote interface, a *-221, "Settings conflict"* error is generated and the frequency is adjusted.

- For the *triggered* burst mode, the sync signal is a TTL "low" while the specified number of cycles is output (for the duration of the burst). After the specified number of cycles has been output, the sync signal goes "high" until the next burst. The sync signal is output from the front-panel *SYNC* terminal.
- For the *external gated* burst mode, the sync signal is a TTL "high" when the output is positive, relative to zero volts (or the dc offset value). The signal is a TTL "low" when the output is negative, relative to zero volts (or the dc offset value). The sync signal is output from the front-panel *SYNC* terminal.

- *Front-Panel Operation*: Press the Freq key and select any valid frequency for the selected function (the present amplitude and offset voltage are used).

- *Remote Interface Operation*:

```
FREQuency {<frequency>|MINimum|MAXimum}
```

  You can also use the `APPLy` command to select the function, frequency, amplitude, and offset of the carrier with a single command.

Burst Count

The *burst count* defines the number of cycles to be output per burst. *Used only in the triggered burst mode (internal or external source).*

> *Certain combinations of burst count and carrier frequency are not allowed. If you attempt to specify a burst count that is not valid, the function generator will automatically adjust the count to the maximum value allowed with the present carrier frequency.*
>
> *Make sure you note the restrictions in "Burst Carrier Frequency" starting on page 85 before setting the burst count.*

- Burst count: **1** to 50,000 cycles, in 1 cycle increments. You can also select an infinite burst count. *The default is 1 cycle.*
- When the *internal* trigger source is selected, the specified number of cycles is output continuously at a rate determined by the *burst rate* setting. The burst rate frequency defines the interval between bursts.
- When the *single* or *external* trigger source is selected, the *burst count* and *burst phase* remain in effect but the *burst rate* is ignored. The specified number of cycles is output each time Single is pressed or when a trigger signal is applied to the *Ext Trig* terminal. The function generator is triggered on the *rising edge* of a TTL pulse.
- When the *external gate* burst source is selected, the burst count, burst rate, burst phase, and burst trigger source are ignored (these parameters are used for the triggered burst mode only).
- *Front-Panel Operation*:

  `4: BURST CNT` **`(MOD MENU)`**

- *Remote Interface Operation*:

  `BM:NCYCles {<# cycles>|INFinity|MINimum|MAXimum}`

Burst Rate

The *burst rate* defines the frequency at which *internally* triggered bursts are generated. The burst rate frequency defines the interval between bursts. *Used only in the internal triggered burst mode.*

Keep in mind that the burst rate is different than the "carrier frequency" which specifies the frequency of the bursted signal.

- Burst rate: 10 mHz to 50 kHz. *The default is 100 Hz.*
- The burst rate setting is used only when *internal* triggering is enabled (triggered mode). The burst rate is ignored when single triggering or external triggering is enabled.
- When the *single* or *external* trigger source is selected (triggered mode), the *burst count* and *burst phase* remain in effect but the *burst rate* is ignored.
- When the *external gate* burst source is selected ("gated" burst mode), the burst count, burst rate, burst phase, and burst trigger source are ignored (these parameters are used for the triggered burst mode only).
- It is possible to specify a burst rate which is too fast for the function generator to output with the specified carrier frequency and burst count. If the burst rate is too high, the function generator will internally adjust it as needed to continuously re-trigger the burst. The adjustment is handled internally by the function generator (the burst rate displayed or queried will be the same as specified).

- *Front-Panel Operation*:

  `5: BURST RATE` **`(MOD MENU)`**

- *Remote Interface Operation*:

  `BM:INTernal:RATE {<`*frequency*`>|MINimum|MAXimum}`

Burst Phase

The *burst phase* defines the starting phase of the burst.

- Burst phase: -360 degrees to +360 degrees, in 0.001 degree increments. *The default is 0 degrees.*
- For *sine*, *square*, *triangle*, and *ramp* waveforms, 0 degrees is the point at which the waveform crosses zero volts (or the dc offset value), in a positive-going direction.
- For *arbitrary* waveforms, 0 degrees is the first data point downloaded to memory.
- When the *external gate* source is selected ("gated" burst mode), the burst count, burst rate, burst phase, and burst trigger source are ignored (these parameters are used for the triggered burst mode only).

- *Front-Panel Operation*:

  `6: BURST PHAS` **`(MOD MENU)`**

- *Remote Interface Operation*:

  `BM:PHASe {<`*degrees*`>|MINimum|MAXimum}`

Frequency-Shift Keying (FSK) Modulation

You can configure the function generator to "shift" its output frequency between two preset values using FSK modulation. The rate at which the output shifts between the two frequencies (called the "carrier frequency" and the "hop frequency") is determined by the internal rate generator or the signal level on the rear-panel *FSK* terminal. *The function generator can produce an FSK waveform using sine, square, triangle, ramp, and arbitrary waveforms.*

For more information on the fundamentals of FSK Modulation, refer to chapter 7, "Tutorial".

To Select FSK Modulation

- The **FSK** annunciator turns on when FSK is enabled.
- Only one modulation mode can be enabled at a time. When you enable FSK, the previous modulation mode is turned off.
- *Front-Panel Operation*: Enable FSK *before* you set up the other modulation parameters. Press Shift FSK to output the waveform using the present settings for frequency, amplitude, and offset voltage.
- *Remote Interface Operation*: To ensure proper operation, you should enable FSK *after* you have set up the other modulation parameters.

```
FSKey:STATe {OFF|ON}
```

FSK Carrier Frequency

- Carrier frequency: 10 mHz to 15 MHz (100 kHz for triangle and ramp). *The default is 1 kHz*. You can use sine, square, ramp, triangle, or arbitrary waveforms for the carrier waveform.
- For *arbitrary waveforms*, the maximum carrier frequency depends on the number of points specified in the waveform. The five built-in arbitrary waveforms can be output at a maximum of 5 MHz.

Number of Arb Points	Minimum Frequency	Maximum Frequency
8 to 8,192 (8k)	10 mHz	5 MHz
8,193 to 12,287 (12k)	10 mHz	2.5 MHz
12,288 to 16,000	10 mHz	200 kHz

- For an externally-controlled FSK waveform, the carrier frequency is output when a "low" TTL level is applied to the *FSK* terminal. The hop frequency is output when a "high" TTL level is applied.
- For FSK, the sync signal is referenced to the FSK "hop" signal (not the carrier). A momentary TTL "high" pulse (> 200 ns) is output on the transition to the "hop" frequency. The signal is output from the front-panel *SYNC* terminal.
- *Front-Panel Operation*: Press the Freq key and select any valid frequency for the selected function (the present amplitude and offset voltage are used).
- *Remote Interface Operation*:

  `FREQuency {<`*frequency*`>|MINimum|MAXimum}`

  You can also use the `APPLy` command to select the function, frequency, amplitude, and offset of the carrier with a single command.

FSK "Hop" Frequency

- Hop frequency: 10 mHz to 15 MHz (100 kHz for triangle and ramp). *The default is 100 Hz.* You can use sine, square, ramp, triangle, or arbitrary waveforms for the hop frequency waveshape.
- For *arbitrary waveforms*, the maximum hop frequency depends on the number of points specified in the waveform. The five built-in arbitrary waveforms can be output at a maximum of 5 MHz.

Number of Arb Points	Minimum Frequency	Maximum Frequency
8 to 8,192 (8k)	10 mHz	5 MHz
8,193 to 12,287 (12k)	10 mHz	2.5 MHz
12,288 to 16,000	10 mHz	200 kHz

- For an externally-controlled FSK waveform, the carrier frequency is output when a "low" TTL level is applied to the *FSK* terminal. The hop frequency is output when a "high" TTL level is applied.
- For FSK, the sync signal is referenced to the FSK "hop" signal (not the carrier). A momentary TTL "high" pulse (> 200 ns) is output on the transition to the "hop" frequency. The signal is output from the front-panel *SYNC* terminal.
- *Front-Panel Operation*: After enabling FSK, press *Recall Menu* to go directly to the FSK FREQ command in the menu.

```
8: FSK FREQ   (MOD MENU)
```

- *Remote Interface Operation*:

```
FSKey:FREQuency {<frequency>|MINimum|MAXimum}
```

FSK Rate

The *FSK rate* is the rate at which the output frequency "shifts" between the carrier frequency and the hop frequency when you select the internal FSK source.

- FSK rate (internal source): 10 mHz to 50 kHz. *The default is 10 Hz*.
- The FSK rate is ignored when the external source is selected.
- *Front-Panel Operation*:

  9: FSK RATE **(MOD MENU)**

- *Remote Interface Operation*:

  FSKey:INTernal:RATE {<*rate in Hz*>|MINimum|MAXimum}

FSK Source

- FSK source: **Internal** or External. *The default is Internal.*
- When the *internal* source is selected, the rate at which the output frequency "shifts" between the carrier frequency and hop frequency is determined by the *FSK rate* specified.
- When the *external* source is selected, the output frequency is determined by the signal level on the rear-panel *FSK* terminal. When a "low" TTL level is present, the carrier frequency is output. When a "high" TTL level is present, the hop frequency is output.
- The maximum external FSK rate is 1 MHz.
- The **Ext** annunciator turns on when the function generator is waiting for an external trigger signal.
- *Front-Panel Operation*:

  10: FSK SRC **(MOD MENU)**

- *Remote Interface Operation*:

  FSKey:SOURce {INTernal|EXTernal}

Frequency Sweep

In the frequency sweep mode, the function generator "steps" from the *start frequency* to the *stop frequency* at a *sweep rate* which you specify. You can sweep up or down in frequency, and with either linear or logarithmic spacing. You can also configure the function generator to output a single sweep (one pass from start frequency to stop frequency) by applying an external trigger. *The function generator can produce a frequency sweep for sine, square, triangle, ramp, or arbitrary waveforms.*

For more information on the fundamentals of frequency sweeps, refer to chapter 7, "Tutorial".

To Select Sweep Modulation

- The **Swp** annunciator turns on when the sweep mode is enabled.
- Only one modulation mode can be enabled at a time. When you enable the sweep mode, the previous modulation mode is turned off.
- *Front-Panel Operation*: Enable the sweep mode *before* you set up the other sweep parameters. Press Shift Sweep to output a sweep using the present settings for amplitude and offset voltage.
- *Remote Interface Operation*: To ensure proper operation, you should enable the sweep mode *after* you have set up the other modulation parameters.

```
SWEep:STATe {OFF|ON}
```

Start Frequency and Stop Frequency

The *start frequency* and *stop frequency* set the upper and lower frequency bounds for the sweep. The function generator begins at the start frequency, sweeps to the stop frequency, and then resets back to the start frequency.

- Start and Stop frequencies: 10 mHz to 15 MHz (100 kHz for triangle and ramp). The sweep is phase continuous over the full frequency range. *The default start frequency is 100 Hz. The default stop frequency is 1 kHz.* The `*RST` command sets the start frequency to 10 mHz (*minimum*) and the stop frequency to 15 MHz (*maximum*).
- For *arbitrary waveforms*, the maximum start or stop frequency depends on the number of points specified in the waveform. The five built-in arbitrary waveforms can be output at a maximum of 5 MHz.

Number of Arb Points	Minimum Frequency	Maximum Frequency
8 to 8,192 (8k)	10 mHz	5 MHz
8,193 to 12,287 (12k)	10 mHz	2.5 MHz
12,288 to 16,000	10 mHz	200 kHz

- To sweep ***up*** in frequency, set the start frequency **<** stop frequency. To sweep ***down*** in frequency, set the start frequency **>** stop frequency.
- The sync signal is a TTL "low" at the start of the sweep (when the start frequency is output) and is a TTL "high" at the end of the sweep (when the stop frequency is output). The signal is output from the front-panel *SYNC* terminal.
- *Front-Panel Operation*: After enabling sweeps, press *Recall Menu* to go directly to the START F command in the menu.

```
1: START F   (SWP MENU)
2: STOP F    (SWP MENU)
```

- *Remote Interface Operation*:

```
FREQuency:STARt {<frequency>|MINimum|MAXimum}
FREQuency:STOP {<frequency>|MINimum|MAXimum}
```

Sweep Time

The *sweep time* specifies the number of seconds required to sweep from the start frequency to the stop frequency. The number of frequency points in the sweep depends on the sweep time you select and is automatically calculated by the function generator.

- Sweep time: 1 ms to 500 seconds. *The default is 1 second.*
- The function generator computes between 2,048 and 4,096 frequency points from the start frequency to the stop frequency.
- *Front-Panel Operation*:

```
3: SWP TIME   (SWP MENU)
```

- *Remote Interface Operation*:

```
SWEep:TIME {<seconds>|MINimum|MAXimum}
```

Sweep Mode

You can sweep with either linear or logarithmic spacing. The output frequency of a *linear* sweep changes linearly during the duration of the sweep. For *logarithmic* spacing, the output frequency changes exponentially during the duration of the sweep.

- Sweep mode: **Linear** or Logarithmic. *The default is Linear.*
- *Front-Panel Operation*:

```
4: SWP MODE   (SWP MENU)
```

- *Remote Interface Operation*:

```
SWEep:SPACing {LINear|LOGarithmic}
```

Sweep Trigger Source

In the *triggered sweep* mode, the function generator outputs a single sweep each time a trigger is received. After one sweep from the start frequency to the stop frequency, the function generator waits for the next trigger while outputting the start frequency.

- Sweep trigger source: **Internal**, Single, or External. *The default is Internal.*
- When the *internal* trigger source is enabled (default), the function generator outputs a continuous sweep at a rate determined by the *sweep rate* specified.
- Pressing [Single] to enable the *single* trigger mode also enables the *external* trigger mode. External triggering is like the front-panel single trigger mode except that you apply a trigger signal to the rear-panel *Ext Trig* terminal. The function generator is triggered on the *rising edge* of a TTL pulse.
- The **Trig** annunciator turns on when the function generator is waiting for a single or external trigger.
- *Front-Panel Operation*: Pressing [Single] to enable the single trigger mode also enables the external trigger mode. Press [Shift] [Internal] to enable the internal trigger mode.
- *Remote Interface Operation*:

  ```
  TRIGger:SOURce {IMMediate|EXTernal|BUS}
  ```

  See "Triggering" on page 98 for more information.

Triggering

Applies only to bursts and frequency sweep. You can issue triggers for bursts and sweeps using *internal* triggering, *single* triggering, or *external* triggering.

- Internal or "automatic" triggering is enabled when you turn on the function generator. In this mode, the function generator outputs continuously when the burst mode or sweep mode is selected.

- Single triggering outputs one burst or initiates one sweep each time you press [Single] from the front-panel. Continue pressing this key to re-trigger the function generator.

- Pressing [Single] to enable the *single* trigger mode also enables the *external* trigger mode. External triggering is like the front-panel single trigger mode except that you apply a trigger signal to the rear-panel *Ext Trig* terminal. The function generator is triggered on the *rising edge* of a TTL pulse.

- The [Single] key is disabled when in remote (the **Rmt** annunciator turns on when in remote) and when a function other than burst or sweep is selected.

Trigger Source Choices

Applies only to burst and sweep. You must specify the source from which the function generator will accept a trigger.

- From the *front panel*, the function generator will accept a single trigger, a hardware trigger from the *Ext Trig* terminal, or continuously output bursts or sweeps using the internal trigger. *At power-on, internal trigger is selected.*
- From the *remote interface*, the function generator will accept a software (bus) trigger, a hardware trigger from the *Ext Trig* terminal, or an immediate internal trigger.
- The trigger source setting is stored in *volatile* memory; the source is set to internal trigger (front panel) or immediate (remote interface) when power has been off or after a remote interface reset.
- To select the trigger source from the remote interface, use the following command. The `APPLy` command automatically sets the trigger source to `IMMediate`.

```
TRIGger:SOURce {IMMediate|EXTernal|BUS}
```

Internal Triggering In this trigger mode, the function generator continuously outputs the burst or sweep (as specified by the *burst rate* or *sweep time*). This is the power-on trigger source for both front-panel and remote interface use.

To select the internal trigger source from the remote interface, use the following command. The `APPLy` command automatically sets the trigger source to `IMMediate`.

```
"TRIGGER:SOURCE IMMEDIATE"
```

Single Triggering In the single trigger mode (front panel only), you can manually trigger the function generator by pressing the front-panel Single key. The function generator outputs one burst or initiates one sweep each time you press the key. The **Trig** annunciator turns on when the function generator is waiting for a single trigger.

The front-panel Single *key is disabled when in remote.*

External Triggering In this trigger mode, the function generator will accept a hardware trigger applied to the *Ext Trig* terminal. The function generator outputs one burst or initiates one sweep each time *Ext Trig* receives the *rising edge* of a TTL pulse.

See also "Ext Trig/FSK/Burst Input Terminal," on page 102.

- *Front-Panel Operation*: The external trigger mode is like the single trigger mode except that you apply the trigger to the *Ext Trig* terminal. Pressing Single to enable the single trigger mode also enables the external trigger mode. The **Trig** annunciator turns on when the function generator is waiting for an external trigger.

  The front-panel Single *key is disabled when in remote.*

- *Remote Interface Operation*:

```
TRIGger:SOURce EXTernal
```

Software (Bus) Triggering The bus trigger mode is available only from the remote interface. This mode is similar to the single trigger mode from the front panel, but you trigger the function generator by sending a bus trigger command. The function generator outputs one burst or initiates one sweep each time the trigger command is received.

- To select the bus trigger source, send the following command.

```
"TRIGGER:SOURCE BUS"
```

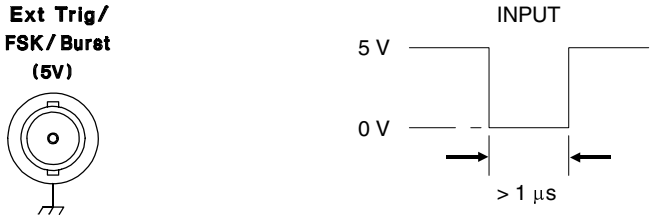
- To trigger the function generator from the remote interface (GPIB or RS-232), send the `*TRG` (trigger) command.
- You can also trigger the function generator from the GPIB interface by sending the IEEE-488 Group Execute Trigger (GET) message. The following statement shows how to send a GET using *BASIC*.

```
TRIGGER 710        Group Execute Trigger
```

- The following statement shows how to send a trigger from the GPIB interface using the GPIB Command Library for *C* or *QuickBASIC*.

```
IOTRIGGER (710)
```

Ext Trig / FSK / Burst Input Terminal



This terminal is used in the following modes:

- *Triggered Sweep Mode*: Press Single or execute `TRIG:SOUR EXT` from the remote interface to enable the triggered sweep mode (sweeps must be enabled). When the *rising edge* of a TTL pulse is received on the *Ext Trig* terminal, the function generator outputs a single sweep. The **Trig** annunciator turns on when the function generator is waiting for a trigger.

- *Externally-Modulated FSK Mode*: To enable the externally-modulated mode, select "FSK SRC EXT" from the front-panel MOD MENU or execute `FSK:SOUR EXT` from the remote interface when FSK is enabled. When a *low* TTL level is present, the FSK carrier frequency is output. When a *high* TTL level is present, the FSK "hop" frequency is output. The **Ext** annunciator turns on when selected.

- *Triggered Burst Mode*: Press Single or execute `TRIG:SOUR EXT` from the remote interface to enable the triggered burst mode (burst must be enabled). When the *rising edge* of a TTL pulse is received, the function generator outputs a burst waveform with the specified number of cycles. The **Trig** annunciator turns on when the function generator is waiting for a trigger.

- *External Gated Burst Mode*: To enable the external gated mode, select "BURST SRC EXT GATED" from the front-panel MOD MENU or execute `BM:SOUR EXT` from the remote interface when burst is enabled. When a *high* TTL level is present, the function generator outputs the carrier waveform. When a *low* TTL level is present, the output is disabled. The **Ext** annunciator turns on when selected.

Arbitrary Waveforms

There are five built-in arbitrary waveforms stored in non-volatile memory. You can also download up to four user-defined waveforms into non-volatile memory. Each waveform can contain between 8 and 16,000 data points.

Refer to chapter 7, "Tutorial", for more information on the internal operation of downloading and outputting an arbitrary waveform.

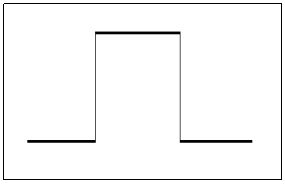
Creating and Storing an Arbitrary Waveform

This section gives an example which shows you how to create and store an arbitrary waveform from the *front panel*. To download an arbitrary waveform from the *remote interface*, see "Arbitrary Waveform Commands" starting on page 174. For this example, you will create and store a pulse waveform containing 400 points.

Shift

Menu On/Off

1 Turn on the menu.



```
A: MOD MENU
```

> >

2 Move across to the EDIT MENU choice on this level.

```
C: EDIT MENU
```

∨

3 Move down a level to the NEW ARB command.

There are seven command choices available in the EDIT MENU. However, you can access only two of the choices (NEW ARB and DELETE) until you either clear volatile memory or load an existing user-defined waveform.

```
1: NEW ARB
```

∨

4 Move down a level to the "PARAMETER" choices.

The first parameter choice is "CLEAR MEM" for the NEW ARB command. If you have previously downloaded any user-defined waveforms, you will also see commands like "GET ARB1" on this level (to load the specified waveform into volatile memory).

```
CLEAR MEM
```

Enter

5 Turn on the waveform edit mode.

The function generator beeps and displays a message to show that the waveform edit mode is now enabled. You are then exited from the menu.

```
LOADING
```

Shift

< Recall Menu

6 Use *Recall Menu* to return to the NEW ARB command.

```
1: NEW ARB
```

>

7 Move across to the POINTS command on this level.

```
2: POINTS
```

∨

8 Move down a level and set the number of points to "400".

```
∧00400 PNTS
```

Enter

9 Save the change and turn off the menu.

The function generator beeps and displays a message to show that the change is now in effect. You are then exited from the menu.

```
COMPUTING
```

Shift

<

Recall Menu

10 Use *Recall Menu* to return to the POINTS command.

```
2: POINTS
```

For this example, the pulse waveform will be "high" for one-half of the cycle (for 200 of the 400 points). From the front panel, each waveform point is a floating-point value between -1 and +1, which correspond to the peak values of the waveform. To define the waveform points, use the LINE EDIT command in the EDIT MENU. The function generator temporarily rescales the waveform amplitude between -1 and +1 at 1 kHz.

>

11 Move across to the LINE EDIT command on this level.

```
3: LINE EDIT
```

v

12 Move down a level and set point "100" to a value of "1".

Use the left/right arrow keys to move between the point field (left) and the floating-point value (right). The comma (,) at the right side of the display indicates that the displayed value is the start point. For this example, the first 99 points in the waveform remain at "0".

```
00100:+1.0000,
```

Enter

13 Press "Enter" to store the start point and move to the end point.

14 Increment to point "299" and set the end point to a value of "1".

Use the left/right arrow keys to move between the point field (left) and the floating-point value (right). As you modify the value, the end point is output. When you press the `Enter` key, the line between the start point and end point is calculated, output, and stored. If you exit without pressing the `Enter` key, the start point and stop point are restored.

```
00299:+1.0000
```

Enter

15 Save the start and stop points and output the waveform.

```
COMPUTING
```

Shift

Menu On/Off

16 Save the waveform and turn off the menu.

The function generator beeps and displays a message to show that the waveform is now stored in volatile memory. You are then exited from the menu.

```
EXITING
```

Shift

< Recall Menu

17 Use *Recall Menu* to return to the LINE EDIT command.

```
3: LINE EDIT
```

> > >

18 Move across to the SAVE AS command on this level.

```
6: SAVE AS
```

[v]

19 Move down a level to save the waveform in non-volatile memory.

At this point, the pulse waveform is being output from volatile memory. You can store up to four user-defined waveforms in non-volatile memory. From the front panel, you can save the waveform using one of the following names: `ARB1`, `ARB2`, `ARB3`, or `ARB4`.

```
ARB1 *NEW*
```

[Enter]

20 Save the waveform and turn off the menu.

The function generator beeps and displays a message to show that the waveform is being saved. You are then exited from the menu.

```
SAVING
```

The waveform is now stored in non-volatile memory and is output from the function generator. The name that you used to store the waveform should appear in the waveform list (press [Arb List]).

Additional Information on Arbitrary Waveforms

- Press the [Arb] key to output the arbitrary waveform currently selected (to scroll through the waveform choices and make a selection, press [Arb List]).

- In addition to creating a new arbitrary waveform from the front panel, you can also edit any existing user-defined waveforms. You can edit waveforms created either from the front panel or from the remote interface. However, you cannot edit any of the five built-in arbitrary waveforms.

  Use the "GET" option under the NEW ARB command in the EDIT MENU to load an existing user-defined waveform. Then, use the LINE EDIT, POINT EDIT, and INVERT commands to edit the waveform.

- The INVERT command in the EDIT MENU takes the inverse of each point in the specified waveform by changing the sign.
- You can use the DELETE command in the EDIT MENU to delete any user-defined waveforms in volatile memory. A list of all user-defined waveforms appears when you move to the DELETE command.

System-Related Operations

This section gives information on topics such as power-down recall, self-test, error conditions, and front-panel display control. This information is not directly related to waveform generation but is an important part of operating the function generator.

Power-Down Recall Mode

When power is turned off, the function generator automatically stores its state in memory location "0". You can configure the function generator to automatically recall the power-down state when power is restored. The recall mode is *disabled* when the function generator is shipped from the factory.

The power-down recall mode can be set from the front-panel only.

- When the recall mode is *disabled* (factory setting), the function generator powers up in its default configuration (see page 123). When the recall mode is *enabled*, the function generator powers up in the power-down configuration.
- The power-down recall setting is stored in *non-volatile* memory, and *does not* change when power has been off or after a remote interface reset.
- The power-down recall mode "remembers" the function (including arbitrary waveforms), frequency, amplitude, dc offset, duty cycle, as well as any modulation parameters.
- Any arbitrary waveforms downloaded to "VOLATILE" memory *are not* remembered. However, if an arbitrary waveform is being output from *non-volatile* memory when the state is stored, the waveform data is stored. The stored waveform is output when the instrument state is recalled.
- *Front-Panel Operation*:

  `2: POWER ON` **`(SYS MENU)`**

  To *enable* the recall mode, select the "LAST STATE" choice.
  To *disable* the recall mode, select the "DEFAULT" choice.

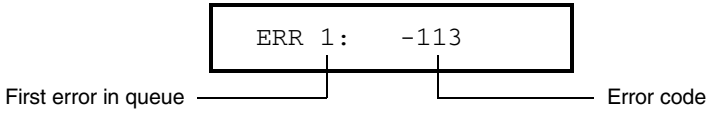
Error Conditions

When the front-panel **ERROR** annunciator turns on, one or more command syntax or hardware errors have been detected. A record of up to 20 errors can be stored in the function generator's *error queue*. *See chapter 5, "Error Messages," for a complete listing of the errors.*

- Errors are retrieved in first-in-first-out (FIFO) order. The first error returned is the first error that was stored. When you have read all errors from the queue, the **ERROR** annunciator turns off. The function generator beeps once each time an error is generated.
- If more than 20 errors have occurred, the last error stored in the queue (the most recent error) is replaced with *-350, "Too many errors"*. No additional errors are stored until you remove errors from the queue. If no errors have occurred when you read the error queue, the function generator responds with *+0, "No error"*.
- The error queue is cleared when power has been off or after a `*CLS` (clear status) command has been executed. The `*RST` (reset) command *does not* clear the error queue.
- *Front-Panel Operation*:

  `3: ERROR` **`(SYS MENU)`**

  If the **ERROR** annunciator is on, press Shift < (Recall Menu) to read the errors stored in the queue. The errors are listed *horizontally* on the "PARAMETER" level. All errors are cleared when you go to the "PARAMETER" level and then turn off the menu.



- *Remote Interface Operation*:

  `SYSTem:ERRor?` *Reads one error from the error queue*

  Errors have the following format (the error string may contain up to *80 characters*):

  ```
  -113,"Undefined header"
  ```

Self-Test

- A *power-on* self-test occurs automatically when you turn on the function generator. This limited test assures you that the function generator is operational.
- A *complete* self-test runs a series of tests and takes approximately 10 seconds to execute. If all tests pass, you can have a high confidence that the function generator is fully operational.
- If the complete self-test is successful, "PASS" is displayed on the front panel. If the self-test fails, "FAIL" is displayed and the **ERROR** annunciator turns on. See the *Service Guide* for instructions on returning the function generator to Agilent for service.
- *Front-Panel Operation*:

  ```
  4: TEST  (SYS MENU)
  ```

  Another way to perform the complete self-test from the front panel is as follows: Hold down Shift as you press the Power switch to turn on the function generator; *hold down the key for more than 5 seconds*. The self-test will begin when you release the key.

- *Remote Interface Operation*:

  ```
  *TST?
  ```

  Returns "0" if the self-test passes or "1" if it fails. If the self-test fails, an error message is also generated with additional information on why the test failed.

Display Control

To speed up the rate at which the function generator can make configuration changes, or for security reasons, you may want to turn off the front-panel display. From the remote interface, you can also display a message containing up to 11 characters on the front panel.

The display can be enabled/disabled from the remote interface only.

- When the display is turned off, output parameters are not sent to the display and all annunciators except **ERROR** and **Shift** are disabled. Front-panel operation is otherwise unaffected by turning off the display.
- The display state is stored in *volatile* memory; the display is always enabled when power has been off or after a remote interface reset.
- You can display a message on the front panel by sending a command from the remote interface. The function generator can display up to 11 characters of the message on the front panel; any additional characters are truncated. Commas, periods, and semicolons share a display space with the preceding character, and *are not* considered individual characters. When a message is displayed, information relating to the output waveform (such as frequency and amplitude) is not sent to the display.
- Sending a message to the display from the remote interface overrides the display state; this means that you can display a message even if the display is turned off.

- The display state is automatically turned on when you return to the local (front panel) state. Press the Shift key or execute `LOCAL 710` from the remote interface to return to the local state.

- *Remote Interface Operation*:

`DISPlay {OFF|ON}`	*Disable/enable the display*
`DISPlay:TEXT` <*quoted string*>	*Display the string enclosed in quotes*
`DISPlay:TEXT:CLEar`	*Clear the displayed message*

The following command string shows how to display a message on the front panel.

```
"DISP:TEXT 'HELLO'"
```

Comma Separators

The function generator can display values on the front panel with or without a comma separator. *This feature is available only from the front panel.*

With comma separator
(factory setting)

Without comma separator

- The display format is stored in *non-volatile* memory, and *does not* change when power has been off or after a remote interface reset. The comma separator is *enabled* when the function generator is shipped from the factory.
- *Front-Panel Operation*:

  5: COMMA **(SYS MENU)**

  See also "To turn off the comma separator," on page 52.

Firmware Revision Query

The function generator has three microprocessors for control of various internal systems. You can query the function generator to determine which revision of firmware is installed for each microprocessor.

- The function generator returns three numbers. The first number is the firmware revision number for the *main generator processor*; the second is for the *input/output processor*; and the third is for the *front-panel processor*.
- *Front-Panel Operation*:

  6: REVISION **(SYS MENU)**

  X.X-X.X-X.X

- *Remote Interface Operation*:

  `*IDN?` *Returns* "HEWLETT-PACKARD,33120A,0,X.X-X.X-X.X"

  Be sure to dimension a string variable with at least 40 characters.

Remote Interface Configuration

This section gives information on configuring the remote interface. For information on programming the function generator over the remote interface, see chapter 4, "Remote Interface Reference," starting on page 125.

GPIB Address

Each device on the GPIB (IEEE-488) interface must have a unique address. You can set the function generator's address to any value between 0 and 30. The current address is displayed momentarily on the front panel when you turn on the function generator. *The address is set to "10" when the function generator is shipped from the factory.*

The GPIB address can be set from the front-panel only.

- The address is stored in *non-volatile* memory, and *does not* change when power has been off or after a remote interface reset.
- Your GPIB bus controller has its own address. Be sure to avoid using the bus controller's address for any instrument on the interface bus. Agilent controllers generally use address "21".
- *Front-Panel Operation*:

  1: HPIB ADDR **(I/O MENU)**

  See also "To set the GPIB address," on page 217.

Remote Interface Selection

The function generator is shipped with both an GPIB (IEEE-488) interface and an RS-232 interface. Only one interface can be enabled at a time. The *GPIB interface* is selected when the function generator is shipped from the factory.

The remote interface can be selected from the front-panel only.

- The interface selection is stored in *non-volatile* memory, and *does not* change when power has been off or after a remote interface reset.
- If you select the GPIB interface, you must select a unique address for the function generator. The current address is displayed momentarily on the front panel when you turn on the function generator.
- Your GPIB bus controller has its own address. Be sure to avoid using the bus controller's address for any instrument on the interface bus.
- If you enable the RS-232 interface, you must select the baud rate and parity to be used. "RS-232" is displayed momentarily on the front panel when you turn on the function generator if you have selected this interface.
- *Front-Panel Operation*:

  2: INTERFACE **(I/O MENU)**

  See also "To select the remote interface," on page 218.

> *Refer to "RS-232 Interface Configuration," starting on page 195 for more information on connecting the function generator to a computer, terminal, or modem over the RS-232 interface.*

Baud Rate Selection (RS-232)

You can select one of six baud rates for RS-232 operation. The rate is set to *9600 baud* when the function generator is shipped from the factory.

The baud rate can be set from the front-panel only.

- Select one of the following: 300, 600, 1200, 2400, 4800, or **9600** baud. *The factory setting is 9600 baud*.
- The baud rate selection is stored in *non-volatile* memory, and *does not* change when power has been off or after a remote interface reset.
- *Front-Panel Operation*:

  `3: BAUD RATE` **`(I/O MENU)`**

  See also "To set the baud rate," on page 219.

Parity Selection (RS-232)

You can select the parity for RS-232 operation. The function generator is configured for *no parity and 8 data bits* when shipped from the factory.

The parity can be set from the front-panel only.

- Select one of the following: **None** (*8 data bits*), Even (*7 data bits*), or Odd (*7 data bits*). When you set the parity, you are indirectly setting the number of data bits.
- The parity selection is stored in *non-volatile* memory, and *does not* change when power has been off or after a remote interface reset.
- To download binary data for arbitrary waveforms over the RS-232 interface, you must select *8 data bits* with *no parity*.
- *Front-Panel Operation*:

  `4: PARITY` **`(I/O MENU)`**

  See also "To set the parity," on page 220.

Programming Language Query

The function generator complies with the rules and conventions of the present version of SCPI (*Standard Commands for Programmable Instruments*). You can verify from the front-panel that the SCPI language is selected.

The interface language can be queried from the front-panel only.

- *Front-Panel Operation*:

  5: LANGUAGE **(I/O MENU)**

SCPI Language Version Query

You can determine the SCPI version with which the function generator is in compliance by sending a query from the remote interface.

The SCPI version can be queried from the remote interface only.

- *Remote Interface Operation*:

  ```
  SYSTem:VERSion?
  ```

  Returns a string in the form "YYYY.V" where the "Y's" represent the year of the version, and the "V" represents a version number for that year (for example, 1993.0).

Calibration Overview

This section gives an overview of the calibration features of the function generator. For a more detailed discussion of the calibration procedures, see chapter 4 in the *Service Guide*.

Calibration Security

This feature allows you to enter a security code to prevent accidental or unauthorized calibrations of the function generator. When you first receive your function generator, it is secured. Before you can calibrate the function generator, you must unsecure it by entering the correct security code.

- *The security code is set to "HP033120" when the function generator is shipped from the factory*. The security code is stored in *non-volatile* memory, and *does not* change when power has been off or after a remote interface reset.
- To secure the function generator from the remote interface, the security code may contain up to 12 alphanumeric characters as shown below. The first character *must be a letter*, but the remaining characters can be letters or numbers. You do not have to use all 12 characters but the first character must always be a letter.

  `A _ _ _ _ _ _ _ _ _ _ _` *(12 characters)*

- To secure the function generator from the remote interface so that it can be unsecured from the front panel, use the eight-character format shown below. The first two characters must be "HP" and the remaining characters must be numbers. Only the last six characters are recognized from the front panel, but all eight characters are required. *To unsecure the function generator from the front panel, omit the "HP" and enter the remaining characters as shown on the following pages.*

  `H P _ _ _ _ _ _` *(8 characters)*

To Unsecure for Calibration You can unsecure the function generator for calibration either from the front panel or over the remote interface. *The function generator is secured when shipped from the factory and the security code is set to "HP033120".*

- *Front-Panel Operation*:

  ```
  1: SECURED   (CAL MENU)
  ```

  If the function generator is secured, you will see the above command when you go into the CAL MENU. (If you move across the "COMMANDS" level in the menu, you will notice that the "CALIBRATE" command is hidden when the function generator is secured.) To unsecure the function generator, go to the "PARAMETER" level of the SECURED command, enter the security code, and then press `Enter`.

  ```
  ∧000000:CODE
  ```

  When you go to the "COMMANDS" level in the CAL MENU again, you will notice that the function generator is unsecured. Notice also that the "CALIBRATE" command is no longer hidden and you can perform a calibration.

  ```
  1: UNSECURED
  ```

- *Remote Interface Operation*:

  ```
  CALibration:SECure:STATe {OFF|ON},<code>
  ```

  To unsecure the function generator, send the above command with the same code used to secure calibration. For example,

  ```
  "CAL:SEC:STAT OFF,HP033120"
  ```

To Secure Against Calibration You can secure the function generator against accidental calibration either from the front panel or over the remote interface. *The function generator is secured when shipped from the factory and the security code is set to "HP033120".*

Be sure to read the security code rules on page 118 before attempting to secure the function generator.

- *Front-Panel Operation*:

  `1: UNSECURED` **`(CAL MENU)`**

  If the function generator is unsecured, you will see the above command when you go into the CAL MENU. To secure the function generator, go to the "PARAMETER" level of the UNSECURED command, enter the security code, and then press `Enter`.

  ```
  ∧000000:CODE
  ```

  When you go to the "COMMANDS" level in the CAL MENU again, you will notice that the function generator is secured. Notice also that the "CALIBRATE" command is now hidden and you cannot perform a calibration.

  ```
  1: SECURED
  ```

- *Remote Interface Operation*:

  `CALibration:SECure:STATe {OFF|ON},<code>`

  To secure the function generator, send the above command with the same code as used to unsecure calibration. For example,

  `"CAL:SEC:STAT ON,HP033120"`

To Change the Security Code To change the security code, you must first unsecure the function generator, and then enter a new code. *Be sure to read the security code rules on page 118 before changing the security code.*

- *Front-Panel Operation*: To change the security code, first make sure that the function generator is unsecured. Select the "PARAMETER" level of the UNSECURED command, enter the new security code, and then press `Enter`. *Changing the code from the front panel also changes the code required from the remote interface.*
- *Remote Interface Operation*:

  ```
  CALibration:SECure:CODE <new code>
  ```

  To change the security code, first unsecure the function generator using the old security code. Then, enter the new code. For example,

  ```
  "CAL:SEC:STAT OFF, HP033120"    Unsecure with old code
  "CAL:SEC:CODE ZZ002133"         Enter new code
  ```

Calibration Count

You can determine the number of times that your function generator has been calibrated. Your function generator was calibrated before it left the factory. When you receive your function generator, read the count to determine its initial value.

- The calibration count is stored in *non-volatile* memory, and *does not* change when power has been off or after a remote interface reset.
- The calibration count increments up to a maximum of 32,767 after which it wraps-around to 0. Since the value increments by one for each calibration point, a complete calibration will increase the value by many counts.
- *Front-Panel Operation*:

  ```
  3: CAL COUNT  (CAL MENU)
  ```

- *Remote Interface Operation*:

  ```
  CALibration:COUNt?
  ```

Calibration Message

You can use the calibration message feature to record calibration information about your function generator. For example, you can store such information as the last calibration date, the next calibration due date, the instrument serial number, or even the name and phone number of the person to contact for a new calibration.

You can record information in the calibration message only from the remote interface. However, you can read the message from either the front-panel menu or the remote interface.

- The calibration message may contain up to 40 characters. However, the function generator can display only 11 characters of the message on the front panel (additional characters are truncated).
- The calibration message is stored in *non-volatile* memory, and *does not* change when power has been off or after a remote interface reset.

- *Front-Panel Operation*:

  `4: MESSAGE` **`(CAL MENU)`** *Read the cal message*

- *Remote Interface Operation*:

  `CALibration:STRing` <*quoted string*> *Store the cal message*

  The following command string shows how to store a calibration message.

```
"CAL:STR 'CAL 12-10-93'"
```

Power-On and Reset State

The parameters marked with a bullet (•) are stored in ***non-volatile*** memory.
The factory settings are shown.

Output Configuration	**Power-On/Reset State**
Function	Sine wave
Frequency	1 kHz
Amplitude (into 50 ohms)	100 mV peak-to-peak
Offset	0.00 Vdc
Output Units	Volts peak-to-peak
Output Termination	50 ohms
Modulation	**Power-On/Reset State**
AM Carrier Waveform	1 kHz Sine wave
AM Modulating Waveform	100 Hz Sine wave
AM Depth	100%
FM Carrier Waveform	1 kHz Sine wave
FM Modulating Waveform	10 Hz Sine wave
FM Peak Frequency Deviation	100 Hz
Burst Carrier Frequency	1 kHz Sine wave
Burst Count	1 cycle
Burst Rate	100 Hz
Burst Starting Phase	0 degrees
FSK Carrier Waveform	1 kHz Sine wave
FSK "Hop" Frequency	100 Hz Sine wave
FSK Rate	10 Hz
Modulation State	Off
Sweep Start / Stop Frequency	100 Hz / 1 kHz
Sweep Time	1 second
Sweep Mode	Linear
System-Related Operations	**Power-On/Reset State**
• Power-Down Recall	• Disabled
Display Mode	On
• Comma Separators	• On
Triggering Operations	**Power-On/Reset State**
Trigger Source	Internal
Input/Output Configuration	**Power-On/Reset State**
• GPIB Address	• 10
• Interface	• GPIB (IEEE-488)
• Baud Rate	• 9600 baud
• Parity	• None (8 data bits)
Calibration	**Power-On/Reset State**
Calibration State	Secured

NOTE: *The power-on state will be different if you have enabled the power-down recall mode.*

See "Power-Down Recall Mode" on page 109 for more information.

For your convenience, this table is duplicated on the rear cover of this manual and on the Quick Reference Card.

4

Remote Interface Reference

Remote Interface Reference

- SCPI Command Summary, *starting on page 127*
- Simplified Programming Overview, *starting on page 136*
- Using the APPLy Command, *starting on page 138*
- Output Configuration Commands, *starting on page 145*
- AM Modulation Commands, *starting on page 154*
- FM Modulation Commands, *starting on page 157*
- Burst Modulation Commands, *starting on page 160*
- Frequency-Shift Keying Commands, *starting on page 167*
- Frequency Sweep Commands, *starting on page 170*
- Arbitrary Waveform Commands, *starting on page 174*
- Triggering, *starting on page 186*
- System-Related Commands, *starting on page 188*
- Calibration Commands, *starting on page 193*
- RS-232 Interface Configuration, *starting on page 195*
- RS-232 Interface Commands, *starting on page 200*
- The SCPI Status Registers, *starting on page 201*
- Status Reporting Commands, *starting on page 209*
- An Introduction to the SCPI Language, *starting on page 211*
- Halting an Output in Progress, *starting on page 216*
- To set the GPIB address, *on page 217*
- To select the remote interface, *on page 218*
- To set the baud rate, *on page 219*
- To set the parity, *on page 220*
- SCPI Conformance Information, *starting on page 221*
- IEEE-488 Conformance Information, *on page 225*

For information on the programming commands for the Phase-Lock Option, refer to the User's and Service Guide included with Option 001.

SCPI

If you are a first-time user of the SCPI language, you may want to refer to these sections to become familiar with the language before attempting to program the function generator.

SCPI Command Summary

This section summarizes the SCPI (*Standard Commands for Programmable Instruments*) commands available to program the function generator over the remote interface. Refer to the later sections in this chapter for more complete details on each command.

Throughout this manual, the following conventions are used for SCPI command syntax.

- Square brackets (**[ ]**) indicate optional keywords or parameters.
- Braces (**{ }**) enclose parameters within a command string.
- Triangle brackets (**< >**) indicate that you must substitute a value for the enclosed parameter.
- A vertical bar (**|**) separates multiple parameter choices.

First-time SCPI users, see page 211.

The APPLy Commands

(*see page 138 for more information*)

```
APPLy:SINusoid [<frequency> [,<amplitude> [,<offset>] ]]

APPLy:SQUare [<frequency> [,<amplitude> [,<offset>] ]]

APPLy:TRIangle [<frequency> [,<amplitude> [,<offset>] ]]

APPLy:RAMP [<frequency> [,<amplitude> [,<offset>] ]]

APPLy:NOISe [<frequency|DEFault>[1] [,<amplitude> [,<offset>] ]]

APPLy:DC [<frequency|DEFault>[1] [,<amplitude|DEFault>[1] [,<offset>] ]]

APPLy:USER [<frequency> [,<amplitude> [,<offset>] ]]

APPLy?
```

[1] *This parameter is ignored for this command but you MUST specify a value or "DEFault".*

Output Configuration Commands

(*see page 145 for more information*)

```
[SOURce:]
  FUNCtion:SHAPe {SINusoid|SQUare|TRIangle|RAMP|NOISe|DC|USER}
  FUNCtion:SHAPe?

[SOURce:]
  FREQuency {<frequency>|MINimum|MAXimum}
  FREQuency? [MINimum|MAXimum]

[SOURce:]
  PULSe:DCYCle {<percent>|MINimum|MAXimum}
  PULSe:DCYCle? [MINimum|MAXimum]

[SOURce:]
  VOLTage {<amplitude>|MINimum|MAXimum}
  VOLTage? [MINimum|MAXimum]
  VOLTage:OFFSet {<offset>|MINimum|MAXimum}
  VOLTage:OFFSet? [MINimum|MAXimum]
  VOLTage:UNIT {VPP|VRMS|DBM|DEFault}
  VOLTage:UNIT?

OUTPut:LOAD {50|INFinity|MINimum|MAXimum}
OUTPut:LOAD? [MINimum|MAXimum]

OUTPut:SYNC {OFF|ON}
OUTPut:SYNC?

*SAV {0|1|2|3}        State 0 is the instrument state at power down.
*RCL {0|1|2|3}        States 1, 2, and 3 are user-defined instrument states.

MEMory:STATe:DELete {0|1|2|3}
```

Default parameters are shown in **bold**.

Modulation Commands

(*see page 154 for more information*)

```
[SOURce:]
  AM:DEPTh {<depth in percent>|MINimum|MAXimum}
  AM:DEPTh? [MINimum|MAXimum]
  AM:INTernal:FUNCtion {SINusoid|SQUare|TRIangle|RAMP|NOISe|USER}
  AM:INTernal:FUNCtion?
  AM:INTernal:FREQuency {<frequency>|MINimum|MAXimum}
  AM:INTernal:FREQuency? [MINimum|MAXimum]
  AM:SOURce {BOTH|EXTernal}
  AM:SOURce?
  AM:STATe {OFF|ON}
  AM:STATe?

[SOURce:]
  FM:DEViation {<peak deviation in Hz>|MINimum|MAXimum}
  FM:DEViation? [MINimum|MAXimum]
  FM:INTernal:FUNCtion {SINusoid|SQUare|TRIangle|RAMP|NOISe|USER}
  FM:INTernal:FUNCtion?
  FM:INTernal:FREQuency {<frequency>|MINimum|MAXimum}
  FM:INTernal:FREQuency? [MINimum|MAXimum]
  FM:STATe {OFF|ON}
  FM:STATe?

[SOURce:]
  BM:NCYCles {<# cycles>|INFinity|MINimum|MAXimum}
  BM:NCYCles? [MINimum|MAXimum]
  BM:PHASe {<degrees>|MINimum|MAXimum}
  BM:PHASe? [MINimum|MAXimum]
  BM:INTernal:RATE {<frequency>|MINimum|MAXimum}
  BM:INTernal:RATE? [MINimum|MAXimum]
  BM:SOURce {INTernal|EXTernal}                    Gated Burst Mode
  BM:SOURce?
  BM:STATe {OFF|ON}
  BM:STATe?

TRIGger:SOURce {IMMediate|EXTernal|BUS}      Triggered Burst Mode
TRIGger:SOURce?
```

Default parameters are shown in **bold**.

Frequency-Shift Keying (FSK) Commands

(*see page 167 for more information*)

```
[SOURce:]
  FSKey:FREQuency {<frequency>|MINimum|MAXimum}
  FSKey:FREQuency? [MINimum|MAXimum]
  FSKey:INTernal:RATE {<rate in Hz>|MINimum|MAXimum}
  FSKey:INTernal:RATE? [MINimum|MAXimum]
  FSKey:SOURce {INTernal|EXTernal}
  FSKey:SOURce?
  FSKey:STATe {OFF|ON}
  FSKey:STATe?
```

Sweep Commands

(*see page 170 for more information*)

```
[SOURce:]
  FREQuency:STARt {<frequency>|MINimum|MAXimum}
  FREQuency:STARt? [MINimum|MAXimum]
  FREQuency:STOP {<frequency>|MINimum|MAXimum}
  FREQuency:STOP? [MINimum|MAXimum]

[SOURce:]
  SWEep:SPACing {LINear|LOGarithmic}
  SWEep:SPACing?
  SWEep:TIME {<seconds>|MINimum|MAXimum}
  SWEep:TIME? [MINimum|MAXimum]
  SWEep:STATe {OFF|ON}
  SWEep:STATe?

TRIGger:SOURce {IMMediate|EXTernal|BUS}      Triggered Sweep Mode
TRIGger:SOURce?
```

Default parameters are shown in **bold**.

Arbitrary Waveform Commands

(*see page 174 for more information*)

```
[SOURce:]
  FUNCtion:USER {<arb name>[1]|VOLATILE}
  FUNCtion:USER?
  FUNCtion:SHAPe USER
  FUNCtion:SHAPe?
```

[1] *Specify 1 of the 5 built-in waveforms or a user-defined waveform name.*

```
DATA VOLATILE, <value>,<value>, . . .

DATA:DAC VOLATILE, {<binary block>|<value>,<value>, . . .}

DATA:ATTRibute:AVERage? [<arb name>]
DATA:ATTRibute:CFACtor? [<arb name>]
DATA:ATTRibute:POINts? [<arb name>]
DATA:ATTRibute:PTPeak? [<arb name>]

DATA:CATalog?

DATA:COPY <destination arb name> [,VOLATILE]

DATA:DELete <arb name>
DATA:DELete:ALL

DATA:NVOLatile:CATalog?
DATA:NVOLatile:FREE?

FORMat:BORDer {NORMal|SWAPped}        Specify Byte Order
FORMat:BORDer?
```

Default parameters are shown in **bold**.

Triggering Commands

(*see page 186 for more information*)

```
TRIGger:SOURce {IMMediate|EXTernal|BUS}
TRIGger:SOURce?

TRIGger:SLOPe {POSitive|NEGative}
TRIGger:SLOPe?

*TRG
```

System-Related Commands

(*see page 188 for more information*)

```
DISPlay {OFF|ON}
DISPlay?

DISPlay:TEXT <quoted string>
DISPlay:TEXT?
DISPlay:TEXT:CLEar

SYSTem:BEEPer

SYSTem:ERRor?

SYSTem:VERSion?

*IDN?

*RST

*TST?

*SAV {0|1|2|3}
*RCL {0|1|2|3}
```

State 0 is the instrument state at power down.
States 1, 2, and 3 are user-defined instrument states.

```
MEMory:STATe:DELete {0|1|2|3}
```

Default parameters are shown in **bold**.

Calibration Commands

(*see page 193 for more information*)

```
CALibration?

CALibration:COUNt?

CALibration:SECure:CODE <new code>

CALibration:SECure:STATe {OFF|ON},<code>
CALibration:SECure:STATe?

CALibration:SETup <0|1|2|3| . . . |84>
CALibration:SETup?

CALibration:STRing <quoted string>
CALibration:STRing?

CALibration:VALue <value>
CALibration:VALue?
```

RS-232 Interface Commands

(*see page 200 for more information*)

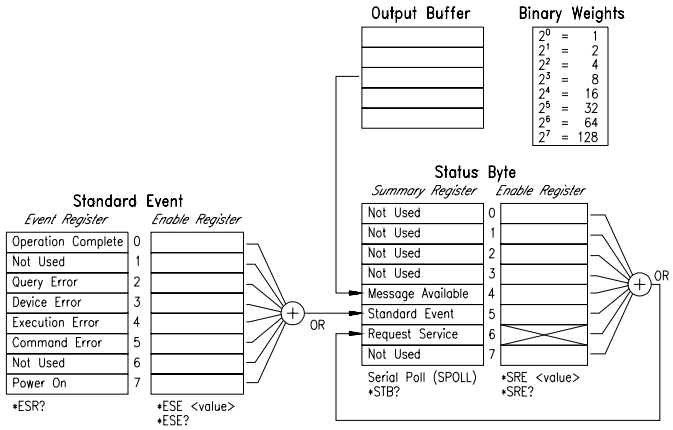
```
SYSTem:LOCal

SYSTem:REMote

SYSTem:RWLock
```

Default parameters are shown in **bold**.

Status Reporting Commands

(*see page 209 for more information*)

```
SYSTem:ERRor?

*CLS

*ESE <enable value>
*ESE?

*ESR?

*OPC

*OPC?

*PSC {0|1}
*PSC?

*SRE <enable value>
*SRE?

*STB?

*WAI
```



Default parameters are shown in **bold**.

IEEE-488.2 Common Commands

(*see page 209 for more information*)

```
*CLS

*ESE <enable value>
*ESE?

*ESR?

*IDN?

*OPC

*OPC?

*PSC {0|1}
*PSC?

*RST

*SAV {0|1|2|3}
*RCL {0|1|2|3}

*SRE <enable value>
*SRE?

*STB?

*TRG

*TST?

*WAI
```

State 0 is the instrument state at power down.
States 1, 2, and 3 are user-defined instrument states.

Default parameters are shown in **bold**.

SCPI

First-time SCPI users, see page 211.

Simplified Programming Overview

This section gives an overview of the basic techniques used to program the function generator over the remote interface. This section is only an overview and does not give all of the details you will need to write your own application programs. Refer to the remainder of this chapter and also chapter 6, "Application Programs", for more details and examples. Also refer to the programming reference manual that came with your computer for details on outputting command strings and entering data.

Using the APPLy Command

The `APPLy` command provides the most straightforward method to program the function generator over the remote interface. For example, the following statement executed from your computer will output a 3 Vpp sine wave at 5 kHz with a -2.5 volt offset:

```
"APPL:SIN 5.0E+3, 3.0, -2.5"
```

Using the Low-Level Commands

Although the `APPLy` command provides the most straightforward method to program the function generator, the low-level commands give you more flexibility to change individual parameters. For example, the following statements executed from your computer will output a 3 Vpp sine wave at 5 kHz with a -2.5 volt offset:

`"FUNC:SHAP SIN"`	*Select sine wave function*
`"FREQ 5.0E+3"`	*Set frequency to 5 kHz*
`"VOLT 3.0"`	*Set amplitude to 3 Vpp*
`"VOLT:OFFS -2.5"`	*Set offset to -2.5 Vdc*

Reading a Query Response

Only the query commands (commands that end with " ? ") will instruct the function generator to send a response message. Queries return either output values or internal instrument settings. For example, the following statements executed from your computer will read the function generator's error queue and print the most recent error:

`dimension statement`	*Dimension string array (80 elements)*
`"SYST:ERR?"`	*Read error queue*
`bus enter statement`	*Enter error string into computer*
`print statement`	*Print error string*

Selecting a Trigger Source

When *burst modulation* or *frequency sweep* is enabled, the function generator will accept an immediate internal trigger, a hardware trigger from the rear-panel *Ext Trig* terminal, or a software (bus) trigger. By default, the internal trigger source is selected. If you want the function generator to use the external source or a bus trigger, you must select that source. For example, the following statements executed from your computer will output a 3-cycle burst each time the *Ext Trig* terminal receives the rising edge of a TTL pulse:

`"BM:NCYC 3"`	*Set burst count to 3 cycles*
`"TRIG:SOUR EXT"`	*Select external trigger source*
`"BM:STAT ON"`	*Enable the burst mode*

Using the APPLy Command

See also "Output Configuration," starting on page 55 in chapter 3.

The APPLy command provides the most straightforward method to program the function generator over the remote interface. You can select the function, frequency, amplitude, and offset all in one command. For example, the following statement outputs a 3 Vpp sine wave at 5 kHz with a -2.5 volt offset using APPLy:

```
"APPL:SIN 5 KHZ, 3.0 VPP, -2.5 V"
```

The syntax statements for the APPLy command are shown on page 143.

Output Frequency

- For the *frequency* parameter of the APPLy command, the output frequency range depends on the function currently selected. You can substitute "MINimum", "MAXimum", or "DEFault" in place of a specific value for the *frequency* parameter. MIN selects the lowest frequency allowed for the selected function. MAX selects the highest frequency allowed. *The default frequency for all functions is 1 kHz.*

Function	Minimum Frequency	Maximum Frequency
Sine	100 μHz	15 MHz
Square	100 μHz	15 MHz
Ramp	100 μHz	100 kHz
Triangle	100 μHz	100 kHz
Built-In Arbs [1]	100 μHz	5 MHz

[1] There are five built-in arbitrary waveforms stored in non-volatile memory: sinc, negative ramp, exponential rise, exponential fall, and cardiac.

- For *arbitrary waveforms* that you create and download to memory, the maximum frequency depends on the number of points specified in the waveform. As shown below, the maximum output frequency decreases as you specify more points in the waveform. The five built-in arbitrary waveforms can be output at a maximum of 5 MHz.

Number of Arb Points	Minimum Frequency	Maximum Frequency
8 to 8,192 (8k)	100 μHz	5 MHz
8,193 to 12,287 (12k)	100 μHz	2.5 MHz
12,288 to 16,000	100 μHz	200 kHz

- *Possible Conflict with Function Change*: The output frequency is automatically adjusted if you select a function whose maximum frequency is *less than* that of the currently active function. For example, if you output a 1 MHz sine wave and then change the function to triangle wave, the function generator will adjust the output to 100 kHz (the upper limit for triangle waves). *From the remote interface, a -221, "Settings conflict" error is generated and the frequency is adjusted.*
- *Possible Conflict with Duty Cycle (square wave only)*: For output frequencies above 5 MHz, the duty cycle is limited to values between 40% and 60% (below 5 MHz, the range is 20% to 80%). The duty cycle is automatically adjusted if you select a frequency that is not valid with the present duty cycle. For example, if you set the duty cycle to 70% and then change the frequency to 8 MHz, the function generator will automatically adjust the duty cycle to 60% (the upper limit for this frequency). *From the remote interface, a -221, "Settings conflict" error is generated and the duty cycle is adjusted.*

Output Amplitude

- For the *amplitude* parameter of the `APPLy` command, the output amplitude range depends on the function currently selected and the output termination. You can substitute "MINimum", "MAXimum", or "DEFault" in place of a specific value for the *amplitude* parameter. MIN selects the smallest amplitude for the selected function (50 mVpp into 50 ohms). MAX selects the largest amplitude (10 Vpp into 50 ohms). *The default amplitude is 100 mVpp (into 50 ohms) for all functions.*

Function	Output Termination	Minimum Amplitude	Maximum Amplitude
Sine	50Ω	50 mVpp	10 Vpp
Square	50Ω	50 mVpp	10 Vpp
Triangle	50Ω	50 mVpp	10 Vpp
Ramp	50Ω	50 mVpp	10 Vpp
Noise	50Ω	50 mVpp	10 Vpp
Built-In Arbs [1]	50Ω	50 mVpp	10 Vpp
Sine	Open Circuit	100 mVpp	20 Vpp
Square	Open Circuit	100 mVpp	20 Vpp
Triangle	Open Circuit	100 mVpp	20 Vpp
Ramp	Open Circuit	100 mVpp	20 Vpp
Noise	Open Circuit	100 mVpp	20 Vpp
Built-In Arbs [1]	Open Circuit	100 mVpp	20 Vpp

[1] There are five built-in arbitrary waveforms stored in non-volatile memory: sinc, negative ramp, exponential rise, exponential fall, and cardiac.

- For *arbitrary waveforms*, the maximum amplitude will be limited if the data points do not span the full range of the output DAC (Digital-to-Analog Converter). For example, the built-in "SINC" waveform does not use the full range of values between ±1 and therefore its maximum amplitude is 6.084 Vpp (into 50 ohms).
- You can set the units for output amplitude to Vpp, Vrms, or dBm. *See "Output Units" on page 64 for more information.*

- *Possible Conflict with Function Change*: The output amplitude is automatically adjusted if you select a function whose maximum amplitude is *less than* that of the currently active function. This conflict may arise when the output units are *Vrms* or *dBm* due to the differences in crest factor for the output functions. For example, if you output a 5 Vrms square wave (into 50 ohms) and then change the function to sine wave, the function generator will adjust the output amplitude to 3.535 Vrms (the upper limit for sine waves in Vrms). *From the remote interface, a -221, "Settings conflict" error is generated and the amplitude is adjusted.*
- *Output Amplitude and Output Termination*: The output amplitude is automatically adjusted (and no error is generated) if you change the output termination. For example, if you set the amplitude to 10 Vpp and then change the termination from 50 ohms to "high impedance", the displayed amplitude will *double* to 20 Vpp. If you change from "high impedance" to 50 ohms, the displayed amplitude will drop in half. *See "Output Termination" on page 65 for more information.*
- A momentary glitch occurs in the output waveform at certain voltages due to output attenuator switching. This positive-going glitch occurs when the output voltage crosses the break-point voltage either from a lower voltage or a higher voltage. The voltages are shown below (for a 0 volt dc offset):

  ```
  252 mVpp, 399 mVpp, 502 mVpp, 796 mVpp, 1 Vpp, 1.59 Vpp,
  2.0 Vpp, 3.17 Vpp, 3.99 Vpp, 6.32 Vpp, 7.96 Vpp
  ```

- The output voltage will momentarily drop to 0 volts at certain voltages due to output relay switching. This occurs when the output voltage crosses the break-point voltage either from a lower voltage or a higher voltage. The voltages are shown below (for a 0 volt dc offset):

  ```
  317 mVpp, 632 mVpp, 1.26 Vpp, 2.52 Vpp, 5.02 Vpp
  ```

DC Offset Voltage

- For the *offset* parameter of the `APPLy` command, you can substitute "MINimum", "MAXimum", or "DEFault" in place of a specific value for the parameter. MIN selects the smallest dc offset voltage for the selected function (0 volts). MAX selects the largest offset for the selected function. *The default offset voltage is 0 volts for all functions.*
- You can set the offset to a positive or negative number with the restrictions shown below. If the specified offset voltage is not valid, the function generator will automatically adjust it to the maximum dc voltage allowed with the present amplitude. (V*max* is the maximum peak-to-peak amplitude for the selected output termination; V*pp* is the output amplitude in volts peak-to-peak.) *From the remote interface, a -221, "Settings conflict" error is generated and the offset is adjusted.*

$$|V_{offset}| + \frac{V_{pp}}{2} \leq V_{max} \quad and \quad |V_{offset}| \leq 2 \times V_{pp}$$

- *DC Offset and Output Termination*: The offset voltage is automatically adjusted (and no error is generated) if you change the output termination. For example, if you set the offset to 100 mVdc and then change the termination from 50 ohms to "high impedance", the displayed offset will *double* to 200 mVdc. If you change from "high impedance" to 50 ohms, the displayed offset will drop in half. *See "Output Termination" on page 65 for more information.*
- For *arbitrary waveforms*, the **Offset** annunciator will turn on if the waveform data has an inherent offset present (if the average is not equal to zero). The function generator calculates the average of the data points and compares the average to zero volts. If the average is not within two DAC (Digital-to-Analog Converter) counts of zero volts, the **Offset** annunciator turns on.
- For *dc volts*, the output level is actually controlled by setting the offset voltage. You can set the dc voltage to any value between ±5 Vdc into 50 ohms or ±10 Vdc into an open circuit.

APPLy Command Syntax

- Because of the use of optional parameters in the APPLy commands (enclosed in square brackets), you must specify *frequency* to use the *amplitude* parameter, and you must specify both *frequency* and *amplitude* to use the *offset* parameter. The following statement is valid (*frequency* and *amplitude* are specified; *offset* is omitted):

  ```
  "APPL:SIN 5.0E+3, 3.0"
  ```

  However, you CANNOT specify *amplitude* and *offset* without *frequency*.

- You can substitute "MINimum", "MAXimum", or "DEFault" in place of specific values for the *frequency*, *amplitude*, and *offset* parameters. For example, the following statement outputs a 3 Vpp sine wave at 15 MHz (the maximum frequency for sine) with a -2.5 volt offset:

  ```
  "APPL:SIN MAX, 3.0, -2.5"
  ```

APPLy:SINusoid [<*frequency*>** [,**<*amplitude*>** [,**<*offset*>**]]]**
Output a sine wave with the specified frequency, amplitude, and dc offset. The waveform is output as soon as the command is executed.

APPLy:SQUare [<*frequency*>** [,**<*amplitude*>** [,**<*offset*>**]]]**
Output a square wave with the specified frequency, amplitude, and dc offset. The waveform is output as soon as the command is executed.

APPLy:TRIangle [<*frequency*>** [,**<*amplitude*>** [,**<*offset*>**]]]**
Output a triangle wave with the specified frequency, amplitude, and dc offset. The waveform is output as soon as the command is executed.

APPLy:RAMP [<*frequency*>** [,**<*amplitude*>** [,**<*offset*>**]]]**
Output a ramp wave with the specified frequency, amplitude, and dc offset. The waveform is output as soon as the command is executed.

APPLy:NOISe [<*frequency*|DEFault> [,<*amplitude*> [,<*offset*>]]]
Output noise with the specified amplitude and dc offset. The waveform is output as soon as the command is executed.

- The *frequency* parameter is ignored for this command but you MUST specify a value or "DEF". If you specify a frequency, the value is remembered when you change to a different function. For example:

```
"APPL:NOIS DEF, 5.0, 2.0"
```

APPLy:DC [<*frequency*|DEFault> [,<*amplitude*|DEFault> [,<*offset*>]]]
Output a dc voltage with the level specified by the *offset* parameter. The dc voltage is output as soon as the command is executed.

- The *frequency* and *amplitude* parameters are ignored for this command but you MUST specify a value or "DEF". If you specify a frequency and amplitude, the values are remembered when you change to a different function. For example:

```
"APPL:DC DEF, DEF, -2.5"
```

APPLy:USER [<*frequency*> [,<*amplitude*> [,<*offset*>]]]
Output the arbitrary waveform currently selected by the FUNC:USER command. The waveform is output using the specified frequency, amplitude, and dc offset. The waveform is output as soon as the command is executed.

See "Arbitrary Waveform Commands" on page 174 for more information on downloading arbitrary waveforms to memory.

APPLy?
Query the function generator's present configuration and return a quoted string. The function, frequency, amplitude, and offset voltage are returned as shown in the sample string below (the quotation marks are returned as part of the string).

```
"SIN +5.000000000000E+03,+3.000000E+00,-2.500000E+00"
```

Output Configuration Commands

See also "Output Configuration," starting on page 55 in chapter 3.

This section describes the low-level commands used to program the function generator. Although the APPLy command provides the most straightforward method to program the function generator, the low-level commands give you more flexibility to change individual parameters.

FUNCtion:SHAPe {SINusoid|SQUare|TRIangle|RAMP|NOISe|DC|USER}
Select the output function. The selected waveform is output using the previously selected frequency, amplitude, and dc offset settings. If you select "USER", the function generator outputs the arbitrary waveform currently selected by the FUNC:USER command. *The default is SIN.*
[*Stored in volatile memory*]

- The following matrix shows which output functions are allowed with each modulation mode. Each "X" indicates a valid combination. If you change to a function that is not allowed with the selected modulation, the modulation mode is turned off.

	Sine	Square	Triangle	Ramp	Noise	Arb
AM Carrier	X	X	X	X		X
AM Modulating Wave	X	X	X	X	X	X
FM Carrier	X	X	X	X		X
FM Modulating Wave	X	X	X	X	X	X
FSK Modulation	X	X	X	X		X
Burst Modulation	X	X	X	X		X
Frequency Sweep	X	X	X	X		X

FUNCtion:SHAPe?
Query the output function. Returns "SIN", "SQU", "TRI", "RAMP", "NOIS", "DC", or "USER".

FREQuency {<*frequency*>|MINimum|MAXimum}
Set the output frequency. MIN selects the lowest frequency allowed for the currently active function. MAX selects the highest frequency allowed for the currently active function. *The default frequency is 1 kHz for all functions*. [*Stored in volatile memory*]

Function	Minimum Frequency	Maximum Frequency
Sine	100 μHz	15 MHz
Square	100 μHz	15 MHz
Ramp	100 μHz	100 kHz
Triangle	100 μHz	100 kHz
Built-In Arbs [1]	100 μHz	5 MHz

[1] There are five built-in arbitrary waveforms stored in non-volatile memory: sinc, negative ramp, exponential rise, exponential fall, and cardiac.

- For *arbitrary waveforms* that you create and download to memory, the maximum frequency depends on the number of points specified in the waveform. As shown below, the maximum output frequency decreases as you specify more points in the waveform. The five built-in arbitrary waveforms can be output at a maximum of 5 MHz.

Number of Arb Points	Minimum Frequency	Maximum Frequency
8 to 8,192 (8k)	100 μHz	5 MHz
8,193 to 12,287 (12k)	100 μHz	2.5 MHz
12,288 to 16,000	100 μHz	200 kHz

- *Possible Conflict with Function Change*: The output frequency is automatically adjusted if you select a function whose maximum frequency is *less than* that of the currently active function. For example, if you output a 1 MHz sine wave and then change the function to triangle wave, the function generator will adjust the output to 100 kHz (the upper limit for triangle waves). *From the remote interface, a -221, "Settings conflict" error is generated and the frequency is adjusted.*

FREQuency? [MINimum|MAXimum]
Query the frequency setting for the function currently active. Returns a value in hertz.

PULSe:DCYCle {<*percent*>|MINimum|MAXimum}
Set the duty cycle in percent for *square waves only*. Duty cycle represents the amount of time per cycle that the square wave is high. *The default is 50%.* [*Stored in volatile memory*]

- Duty cycle: 20% to 80%, in 1% increments (*frequency* ≤ 5 MHz).
  40% to 60%, in 1% increments (*frequency* > 5 MHz).
  The default is 50%.
- The duty cycle setting is remembered when you change from square wave to another function. When you return to the square wave function, the previous duty cycle is used. The APPLy command automatically sets the duty cycle to 50% for square waves.
- *Possible Conflict with Output Frequency*: The duty cycle is automatically adjusted if you select a frequency that is not valid with the present duty cycle. For example, if you set the duty cycle to 70% and then change the frequency to 8 MHz, the function generator will automatically adjust the duty cycle to 60% (the upper limit for this frequency). *From the remote interface, a -221, "Settings conflict" error is generated and the duty cycle is adjusted.*

PULSe:DCYCle? [MINimum|MAXimum]
Query the duty cycle setting. Returns a value in percent.

`VOLTage {<`*amplitude***`>|MINimum|MAXimum}`**
Set the output amplitude for the currently active function. MIN selects the smallest amplitude allowed for the selected function (50 mVpp into 50 ohms). MAX selects the largest amplitude allowed (10 Vpp into 50 ohms). *The default amplitude is 100 mVpp (into 50 ohms).*
[*Stored in volatile memory*]

Function	Output Termination	Minimum Amplitude	Maximum Amplitude
Sine	50Ω	50 mVpp	10 Vpp
Square	50Ω	50 mVpp	10 Vpp
Triangle	50Ω	50 mVpp	10 Vpp
Ramp	50Ω	50 mVpp	10 Vpp
Noise	50Ω	50 mVpp	10 Vpp
Built-In Arbs [1]	50Ω	50 mVpp	10 Vpp
Sine	Open Circuit	100 mVpp	20 Vpp
Square	Open Circuit	100 mVpp	20 Vpp
Triangle	Open Circuit	100 mVpp	20 Vpp
Ramp	Open Circuit	100 mVpp	20 Vpp
Noise	Open Circuit	100 mVpp	20 Vpp
Built-In Arbs [1]	Open Circuit	100 mVpp	20 Vpp

[1] There are five built-in arbitrary waveforms stored in non-volatile memory: sinc, negative ramp, exponential rise, exponential fall, and cardiac.

- For *arbitrary waveforms*, the maximum amplitude will be limited if the data points do not span the full range of the output DAC (Digital-to-Analog Converter). For example, the built-in "SINC" waveform does not use the full range of values between ±1 and therefore its maximum amplitude is 6.084 Vpp (into 50 ohms).
- You can set the units for output amplitude to Vpp, Vrms, or dBm. *See the* `VOLT:UNIT` *command for more information.*
- For *dc volts*, the output level is actually controlled by setting the offset voltage. You can set the dc voltage to any value between ±5 Vdc into 50 ohms or ±10 Vdc into an open circuit. *See the* `VOLT:OFFS` *command for more information.*

- *Possible Conflict with Function Change*: The output amplitude is automatically adjusted if you select a function whose maximum amplitude is *less than* that of the currently active function. This conflict may arise when the output units are *Vrms* or *dBm* due to the differences in crest factor for the output functions. For example, if you output a 5 Vrms square wave (into 50 ohms) and then change the function to sine wave, the function generator will adjust the output amplitude to 3.535 Vrms (the upper limit for sine waves in Vrms). *From the remote interface, a -221, "Settings conflict" error is generated and the amplitude is adjusted.*
- *Output Amplitude and Output Termination*: The output amplitude is automatically adjusted (and no error is generated) if you change the output termination. For example, if you set the amplitude to 10 Vpp and then change the termination from 50 ohms to "high impedance", the displayed amplitude will *double* to 20 Vpp. If you change from "high impedance" to 50 ohms, the displayed amplitude will drop in half. *See the* `OUTP:LOAD` *command for more information.*

`VOLTage? [MINimum|MAXimum]`
Query the output amplitude for the currently selected function. Returns the magnitude of the output amplitude. Units are not returned but are in the units set by the most recent `VOLT:UNIT` command.

`VOLTage:OFFSet {<`*offset*`>|MINimum|MAXimum}`
Set the dc offset voltage for the currently active function. MIN selects the smallest dc offset voltage for the selected function (0 volts). MAX selects the largest offset for the selected function. *The default offset voltage is 0 volts for all functions.* [*Stored in volatile memory*]

- You can set the offset to a positive or negative number with the restrictions shown below. If the specified offset voltage is not valid, the function generator will automatically adjust it to the maximum dc voltage allowed with the present amplitude. (V*max* is the maximum peak-to-peak amplitude for the selected output termination; V*pp* is the output amplitude in volts peak-to-peak.) *From the remote interface, a -221, "Settings conflict" error is generated and the offset is adjusted.*

$$|V_{offset}| + \frac{V_{pp}}{2} \leq V_{max} \quad and \quad |V_{offset}| \leq 2 \times V_{pp}$$

VOLTage:OFFSet *(continued)*

- *DC Offset and Output Termination*: The offset voltage is automatically adjusted (and no error is generated) if you change the output termination. For example, if you set the offset to 100 mVdc and then change the termination from 50 ohms to "high impedance", the displayed offset will *double* to 200 mVdc. If you change from "high impedance" to 50 ohms, the displayed offset will drop in half. *See the* `OUTP:LOAD` *command for more information.*
- For *dc volts*, the output level is actually controlled by setting the offset voltage. You can set the dc voltage to any value between ±5 Vdc into 50 ohms or ±10 Vdc into an open circuit.
- For *arbitrary waveforms*, the **Offset** annunciator will turn on if the waveform data has an inherent offset present (if the average is not equal to zero). The function generator calculates the average of the data points and compares the average to zero volts. If the average is not within two DAC (Digital-to-Analog Converter) counts of zero volts, the **Offset** annunciator turns on.

`VOLTage:OFFSet? [MINimum|MAXimum]`
Query the dc offset voltage for the currently selected function. Returns a value in dc volts.

`VOLTage:UNIT {VPP|VRMS|DBM|DEFault}`
Select the output units for *amplitude only* (*does not affect offset*). When switching from remote programming over the interface back to front-panel (local) operation, the output units are remembered. For example, if you select "Vrms" from the remote interface, the units are shown as "Vrms" on the front-panel display. *The default is VPP.*
[*Stored in volatile memory*]

`VOLTage:UNIT?`
Query the units selected. Returns "VPP", "VRMS", or "DBM".

OUTPut:LOAD {50|INFinity|MINimum|MAXimum}
Select the output termination for output amplitude and offset voltage. The function generator has a fixed output impedance of 50 ohms on the *OUTPUT* terminal. You can specify whether **you** are terminating the output into a 50 ohm load or an open circuit. Incorrect impedance matching between the function generator and your load will result in an amplitude or offset which does not match the specified signal level.
[*Stored in volatile memory*]

- INF sets the output termination to "high impedance". MIN selects 50Ω. MAX selects "high impedance". *The default is "50".*
- The amplitude (or dc offset) is automatically adjusted (and no error is generated) if you change the output termination. For example, if you set the amplitude to 10 Vpp and then change the termination from 50 ohms to "high impedance", the displayed amplitude will *double* to 20 Vpp. If you change from "high impedance" to 50 ohms, the displayed amplitude will drop in half.
- If you specify a 50 ohm termination but are actually terminating into an open circuit, the displayed output will be *twice* the value specified. For example, if you set the offset to 100 mVdc (and specify a 50 ohm termination) but are actually terminating the output into an open circuit, the actual displayed offset will be 200 mVdc.

OUTPut:LOAD? [MINimum|MAXimum]
Query the output impedance. Returns "50" or "9.9E+37".

OUTPut:SYNC {OFF|ON}
Disable or enable output from the *SYNC* terminal. *The default is "ON".* When the sync signal is disabled, the output level on the *SYNC* terminal is indeterminate (it might be a TTL "high" or a TTL "low").
[*Stored in volatile memory*]

See "SYNC Signal" on page 68 in chapter 3 for more information on the SYNC signal provided for each of the output functions.

OUTPut:SYNC?
Query the state of the *SYNC* terminal. Returns "0" (OFF) or "1" (ON).

***SAV {0|1|2|3}**
Store up to four different instrument configurations.
[*Stored in non-volatile memory*]

- Four memory locations (numbered 0, 1, 2, and 3) are available to store instrument configurations. The state storage feature "remembers" the function (including arbitrary waveforms), frequency, amplitude, dc offset, duty cycle, as well as any modulation parameters. To recall a stored state, you must use the same memory location used previously to store the state.
- From the remote interface only, you can use memory location "0" to store a fourth instrument configuration (you cannot store to this memory location from the front panel). However, keep in mind that memory location "0" is automatically overwritten when the power is turned off.
- Any arbitrary waveforms downloaded to "VOLATILE" memory *are not* remembered. However, if an arbitrary waveform is being output from *non-volatile* memory when the state is stored, the waveform data is stored. The stored waveform is output when the instrument state is recalled.
- If you delete an arbitrary waveform after storing the state, the waveform data is lost and the function generator will not output the waveform when the state is recalled. The "SINC" waveform is output in place of the deleted waveform.
- When power is turned off, the function generator automatically stores its state in memory location "0". You can configure the function generator to automatically recall the power-down state when power is restored. The recall mode is *disabled* when the function generator is shipped from the factory.

  Select the POWER ON LAST STATE command from the SYS MENU to enable the power-down recall mode. Select POWER ON DEFAULT to disable the power-down recall mode. *See "Power-Down Recall Mode" on page 109 for more information.*

`*RCL {0|1|2|3}`
Recall a previously stored state. To recall a stored state, you must use the same memory location used previously to store the state.

- You cannot recall the instrument state from a memory location that was not previously specified as a storage location. For example, an error is generated if you attempt to recall from memory location "2" but have never stored to that location. *From the remote interface, a +810, "State has not been stored" error is generated if nothing is stored in the specified memory location.*

`MEMory:STATe:DELete {0|1|2|3}`
Delete a previously stored state and clear the memory location.
If nothing is stored in the specified memory location, a *+810, "State has not been stored"* error is generated.

AM Modulation Commands

See also "Amplitude Modulation," starting on page 71 in chapter 3.

AM Overview

The following is an overview of the steps required to generate an AM waveform. The commands used for AM are listed on the next page.

1 Set up the carrier waveform.

Use the `APPLy` command or the equivalent `FUNC:SHAP`, `FREQ`, `VOLT`, and `VOLT:OFFS` commands to select the function, frequency, amplitude, and offset of the carrier waveform. You can select a sine, square, triangle, ramp, or arbitrary waveform for the carrier.

2 Select the shape of the modulating waveform.

You can modulate the carrier with a sine, square, triangle, ramp, noise, or arbitrary waveform. Use the `AM:INT:FUNC` command to select the modulating waveshape.

3 Set the modulating frequency.

Set the modulating frequency to any value between 10 mHz and 20 kHz using the `AM:INT:FREQ` command.

4 Set the modulation depth.

Set the modulation depth to a value between 0% and 120% using the `AM:DEPT` command.

5 Select the modulation source.

The function generator will accept an internal modulating signal, an external modulating signal, or both. Select the modulating source using the `AM:SOUR` command.

6 Enable AM modulation.

After you have set up the other modulation parameters, use the `AM:STAT ON` command to enable AM.

AM Commands

> *Use the* APPLy *command or the equivalent* FUNC:SHAP, FREQ, VOLT, *and* VOLT:OFFS *commands to configure the carrier waveform. Set the carrier frequency between* 100 μHz *and* 15 MHz *(100 kHz for triangle and ramp). The default is 1 kHz.*

AM:DEPTh {<*depth in percent***>|MINimum|MAXimum}**
Set the internal modulation depth in percent. Select from 0% to 120%. *The default is 100%*. MIN = 0%. MAX = 120%. [*Stored in volatile memory*]

If you select the *external* modulating source (AM:SOUR EXT), the modulation depth is controlled by the signal level present on the *AM Modulation* terminal (5 volts **peak** corresponds to 100% modulation).

AM:DEPTh? [MINimum|MAXimum]
Query the modulation depth. Returns a value in percent.

AM:INTernal:FUNCtion {SINusoid|SQUare|TRIangle|RAMP|NOISe|USER}
Select the shape of the *modulating* waveform. Used only when the *internal* modulation source is selected (AM:SOUR INT). You can use the noise function as the modulating waveform. However, you *cannot* use the noise function or dc volts as the carrier waveform. *The default is SIN*. [*Stored in volatile memory*]

AM:INTernal:FUNCtion?
Query the shape of the internal modulating waveform. Returns “SIN”, “SQU”, “TRI”, “RAMP”, “NOIS”, or “USER”.

AM:INTernal:FREQuency {<*frequency***>|MINimum|MAXimum}**
Set the frequency of the *modulating* waveform. Used only when the *internal* modulation source is selected (AM:SOUR INT). Select from 10 mHz to 20 kHz. *The default is 100 Hz*. MIN = 10 mHz. MAX = 20 kHz. [*Stored in volatile memory*]

AM:INTernal:FREQuency? [MINimum|MAXimum]
Query the internal modulating frequency. Returns a value in hertz.

AM:SOURce {BOTH|EXTernal}
Select the source of the modulating signal. The function generator will accept an internal modulating signal, an external modulating signal, or both. *The default is BOTH*. [*Stored in volatile memory*]

- ***The External modulating source is always enabled.***
- When both sources are enabled (*internal-external*), the function generator adds the internal and external modulating signals (the carrier waveform is actually modulated with two waveforms).
- When the internal source is disabled (*external only*), the carrier waveform is modulated with the external waveform. The **Ext** annunciator turns on to indicate that the function generator is expecting a modulating signal on the rear-panel *AM Modulation* terminal.
- You apply the external modulating waveform to the *AM Modulation* terminal. The *modulation depth* is controlled by the signal level present (5 volts **peak** corresponds to 100% modulation).

AM:SOURce?
Query the modulating source. Returns "BOTH" or "EXT".

AM:STATe {OFF|ON}
Disable or enable AM. To ensure proper operation, you should enable AM *after* you have set up the other modulation parameters. Only one modulation mode can be enabled at a time. When you enable AM, the previous modulation mode is turned off.

AM:STATe?
Query the state of AM. Returns "0" (OFF) or "1" (ON).

FM Modulation Commands

See also "Frequency Modulation," starting on page 76 in chapter 3.

FM Overview

The following is an overview of the steps required to generate an FM waveform. The commands used for FM are listed on the next page.

1 Set up the carrier waveform.

Use the `APPLy` command or the equivalent `FUNC:SHAP`, `FREQ`, `VOLT`, and `VOLT:OFFS` commands to select the function, frequency, amplitude, and offset of the carrier waveform. You can select a sine, square, triangle, ramp, or arbitrary waveform for the carrier.

2 Select the shape of the modulating waveform.

You can modulate the carrier with a sine, square, triangle, ramp, noise, or arbitrary waveform. Use the `FM:INT:FUNC` command to select the modulating waveshape.

3 Set the modulating frequency.

Set the modulating frequency to any value between 10 mHz and 10 kHz using the `FM:INT:FREQ` command.

4 Set the peak frequency deviation.

Set the deviation to a value between 10 mHz and 7.5 MHz using the `FM:DEV` command.

5 Enable FM modulation.

After you have set up the other modulation parameters, use the `FM:STAT ON` command to enable FM.

FM Commands

> *Use the* APPLy *command or the equivalent* FUNC:SHAP, FREQ, VOLT, *and* VOLT:OFFS *commands to configure the carrier waveform. Set the carrier frequency between 10 mHz and 15 MHz (100 kHz for triangle and ramp). The default is 1 kHz.*

FM:DEViation {<*peak deviation in Hz***>|MINimum|MAXimum}**
Set the peak frequency deviation in hertz. This value represents the variation in frequency of the modulating waveform from the carrier frequency. Select from 10 mHz to 7.5 MHz. *The default is 100 Hz*. MIN = 10 mHz. MAX = 7.5 MHz. [*Stored in volatile memory*]

- The *carrier frequency* must always be greater than or equal to the peak frequency deviation. If you attempt to set the deviation to a value greater than the carrier frequency (with FM enabled), the function generator will automatically adjust the deviation to equal the present carrier frequency. *From the remote interface, a -221, "Settings conflict" error is generated and the deviation is adjusted.*
- The sum of the *carrier frequency* and peak frequency deviation must be less than or equal to the maximum frequency for the selected function ***plus 100 kHz*** (15.1 MHz for sine and square, 200 kHz for triangle and ramp, and 5.1 MHz for arbitrary waveforms). If you attempt to set the deviation to a value that is not valid, the function generator will automatically adjust the deviation to the maximum value allowed with the present carrier frequency. *From the remote interface, a -221, "Settings conflict" error is generated and the deviation is adjusted.*

FM:DEViation? [MINimum|MAXimum]
Query the peak frequency deviation. Returns a value in hertz.

FM:INTernal:FUNCtion {SINusoid|SQUare|TRIangle|RAMP|NOISe|USER}
Select the shape of the *modulating* waveform. You can use the noise function as the modulating waveform. However, you *cannot* use the noise function or dc volts as the carrier waveform. *The default is SIN.*
[*Stored in volatile memory*]

FM:INTernal:FUNCtion?
Query the shape of the modulating waveform. Returns "SIN", "SQU", "TRI", "RAMP", "NOIS", or "USER".

FM:INTernal:FREQuency {<*frequency*>|MINimum|MAXimum}
Set the frequency of the *modulating* waveform. Select from 10 mHz to 10 kHz. *The default is 10 Hz*. MIN = 10 mHz. MAX = 10 kHz.
[*Stored in volatile memory*]

FM:INTernal:FREQuency? [MINimum|MAXimum]
Query the modulating frequency. Returns a value in hertz.

FM:STATe {OFF|ON}
Disable or enable FM. To ensure proper operation, you should enable FM *after* you have set up the other modulation parameters. Only one modulation mode can be enabled at a time. When you enable FM, the previous modulation mode is turned off.

FM:STATe?
Query the state of FM. Returns "0" (OFF) or "1" (ON).

Burst Modulation Commands

See also "Burst Modulation," starting on page 81 in chapter 3.

Burst Modulation Overview

The following is an overview of the steps required to generate a burst-modulated waveform. *The commands used for burst modulation are listed on page 162.*

You can use burst modulation in the following two modes. The function generator enables one burst mode at a time based on the *trigger source* and *burst source* that you select (see the table below).

- *Triggered Burst Mode*: In this mode, the function generator outputs a waveform with a specified number of cycles (*burst count*) each time a trigger is received. After the specified number of cycles has been output, the function generator waits for the next trigger. You can configure the function generator to accept an internal trigger or you can send the trigger externally by pressing the front-panel Single key or by applying a trigger signal to the rear-panel *Ext Trig* terminal.

  At power-on, the internally-triggered burst mode is enabled.

- *External Gated Burst Mode*: In this mode, the output waveform is either "on" or "off" based on the level of the external signal applied to the rear-panel *Ext Trig* terminal. When the gate signal is true, the function generator outputs a continuous waveform. When the gate signal is false, the output is turned off.

	Trigger Source (TRIG:SOUR)	**Burst Source** (BM:SOUR)	**Burst Count** (BM:NCYC)	**Burst Rate** (BM:INT:RATE)	**Burst Phase** (BM:PHAS)
External Triggered Mode	External	Internal	Available	Not Used	Available
Internal Triggered Mode	Internal	Internal	Available	Available	Available
External Gated Mode	Internal	External	Not Used	Not Used	Not Used

1 Set up the burst carrier waveform.

Use the APPLy command or the equivalent FUNC:SHAP, FREQ, VOLT, and VOLT:OFFS commands to select the function, frequency, amplitude, and offset of the carrier waveform. You can select a sine, square, triangle, ramp, or arbitrary waveform for the carrier.

2 Set the burst count.

Set the burst count (number of cycles per burst) to any value between 1 and 50,000 cycles using the BM:NCYC command. *Used only in the triggered burst mode (internal or external source).*

3 Set the burst rate.

Set the burst rate (the frequency at which *internally* triggered bursts are generated) to any value between 10 mHz and 50 kHz using the BM:INT:RATE command. *Used only in the triggered burst mode with an internal trigger source.*

4 Set the burst starting phase.

Set the starting phase of the burst to any value between -360 degrees and +360 degrees using the BM:PHAS command. *Used only in the triggered burst mode (internal or external source).*

5 Select the trigger source or burst source.

- If you are using the *triggered burst* mode, select the trigger source using the TRIG:SOUR command.
- If you are using the *external gated burst* mode, select the external gate source using the BM:SOUR EXT command.

6 Enable burst modulation.

After you have set up the other modulation parameters, use the BM:STAT ON command to enable the burst mode.

Burst Modulation Commands

> *Use the* APPLy *command or the equivalent* FUNC:SHAP, FREQ, VOLT, *and* VOLT:OFFS *commands to configure the carrier waveform. Set the carrier frequency between 10 mHz and 5 MHz (100 kHz for triangle and ramp). The default is 1 kHz.*

BM:NCYCles {<# *cycles***>|INFinity|MINimum|MAXimum}**
Set the number of cycles to be output per burst (triggered burst mode only). Select from 1 cycle to 50,000 cycles, in 1 cycle increments. *The default is 1 cycle*. MIN = 1 cycle. MAX = 50,000 cycles. [*Stored in volatile memory*]

- For *sine, square,* and *arbitrary* waveforms (does not apply to ramp and triangle waveforms), the relationship between the carrier frequency and the minimum burst count is shown below.

Carrier Frequency	Minimum Burst Count
10 mHz to 1 MHz	1
>1 MHz to 2 MHz	2
>2 MHz to 3 MHz	3
>3 MHz to 4 MHz	4
>4 MHz to 5 MHz	5

For sine, square, and arbitrary waveforms only.

If you attempt to set the carrier frequency to a value that is not valid, the function generator will automatically adjust the frequency to the maximum value allowed with the present burst count. *From the remote interface, a -221, "Settings conflict" error is generated and the carrier frequency is adjusted.*

- For *all waveforms* used with burst, if the carrier frequency is set less than or equal to *100 Hz*, the following relationship applies.

$$\frac{\text{Burst Count}}{\text{Carrier Frequency}} \leq 500 \text{ seconds} \qquad \textit{For Carrier} \leq \textit{100 Hz}$$

  If you attempt to set the carrier frequency to a value that is not valid, the function generator will automatically adjust the frequency to the minimum value allowed with the present burst count. *From the remote interface, a -221, "Settings conflict" error is generated and the frequency is adjusted.*

BM:NCYCles? [MINimum|MAXimum]
Query the burst count. Returns an integer between 1 and 50,000.

BM:PHASe {<*degrees***>|MINimum|MAXimum}**
Set the starting phase for the burst (triggered burst mode only). Select from -360 degrees to +360 degrees, in 0.001 degree increments. *The default is 0 degrees*. MIN = -360 degrees. MAX = +360 degrees. [*Stored in volatile memory*]

- For *sine*, *square*, *triangle*, and *ramp* waveforms, 0 degrees is the point at which the waveform crosses zero volts (or the dc offset value), in a positive-going direction. For *arbitrary* waveforms, 0 degrees is the first point downloaded to memory.

BM:PHASe? [MINimum|MAXimum]
Query the starting phase. Returns a value in degrees.

BM:INTernal:RATE {<*frequency*>|MINimum|MAXimum}
Set the burst rate for internally triggered bursts. The burst rate frequency defines the interval between bursts. Select from 10 mHz to 50 kHz. *The default is 100 Hz*. MIN = 10 mHz. MAX = 50 kHz.
[*Stored in volatile memory*]

- The burst rate setting is used only when *internal* triggering is enabled (triggered mode). The burst rate is ignored when single triggering or external triggering is enabled.
- It is possible to specify a burst rate which is too fast for the function generator to output with the specified carrier frequency and burst count. If the burst rate is too high, the function generator will internally adjust it as needed to continuously re-trigger the burst. The adjustment is handled internally by the function generator (the burst rate displayed or queried will be the same as specified).

BM:INTernal:RATE? [MINimum|MAXimum]
Query the burst rate. Returns a value in hertz.

BM:SOURce {INTernal|EXTernal}
Select the burst modulation source. In the *external gated burst* mode, the output waveform is either "on" or "off" based on the level of the external signal applied to the rear-panel *Ext Trig* terminal. *The default is INT*. [*Stored in volatile memory*]

- When the *internal* burst source is selected, the external gated mode is disabled.
- When the *external gate* source is selected, the output is enabled or disabled based on the logic level of the gate signal applied to the *Ext Trig* terminal. When the gate signal is true (TTL high), the function generator outputs a continuous waveform. When the gate signal is false (TTL low), the output is turned off (zero volts or the dc offset level).
- When the *external gate* source is selected, the burst count, burst rate, burst phase, and burst trigger source are ignored (these parameters are used for the triggered burst mode only).

BM:SOURce?
Query the present burst modulation source. Returns "INT" or "EXT".

BM:STATe {OFF|ON}
Disable or enable burst modulation. To ensure proper operation, you should enable the burst mode *after* you have set up the other modulation parameters. Only one modulation mode can be enabled at a time. When you enable the burst mode, the previous modulation mode is turned off.

BM:STATe?
Query the state of burst modulation. Returns "0" (OFF) or "1" (ON).

TRIGger:SOURce {IMMediate|EXTernal|BUS}
Select the burst "trigger" source. In the *triggered burst* mode, the function generator outputs a waveform with the specified number of cycles (*burst count*) each time a trigger is received. After the specified number of cycles has been output, the function generator waits for the next trigger while outputting no signal (zero volts or the dc offset level). *The default is IMM*. [*Stored in volatile memory*]

- In the triggered burst mode, the "burst source" is *Internal*.
- When the *Immediate* (internal) trigger source is selected, the frequency at which the burst is generated is determined by the *burst rate* (`BM:INT:RATE`). The `APPLy` command automatically sets the trigger source to `IMMediate`.
- When the *External* trigger source is selected, the function generator will accept a hardware trigger applied to the rear-panel *Ext Trig* terminal. The function generator outputs the specified number of cycles each time *Ext Trig* receives the rising edge of a TTL pulse. The **Trig** annunciator turns on when the function generator is waiting for an external trigger.
- When the *Bus* (software) source is selected, the function generator outputs one burst each time a bus trigger command is received. To trigger the function generator from the remote interface (GPIB or RS-232), send the `*TRG` (trigger) command. *You can also trigger the function generator from the GPIB interface by sending the IEEE-488 Group Execute Trigger (GET) message* (e.g., `TRIGGER 710`).
- When the *External* or *Bus* trigger source is selected, the *burst count* and *burst phase* remain in effect but the *burst rate* is ignored.

TRIGger:SOURce *(continued)*

- When the *External gate* source is selected ("gated" burst mode), the specified trigger source is ignored. The external gated mode overrides the triggered mode source.
- To ensure synchronization when the *Bus* source is selected, send the `*WAI` (wait) command. When the `*WAI` command is executed, the function generator waits for all pending operations to complete before executing any additional commands. For example, the following command string guarantees that the first trigger is accepted and executed before the second trigger is recognized.

```
"TRIG:SOUR BUS;*TRG;*WAI;*TRG;*WAI"
```

- You can use the `*OPC?` (operation complete query) command or the `*OPC` (operation complete) command to signal when the burst is complete. The `*OPC?` command returns "1" to the output buffer when the burst is complete. The `*OPC` command sets the "operation complete" bit (bit 0) in the Standard Event register when the burst is complete.

`TRIGger:SOURce?`
Query the present trigger source. Returns "IMM", "EXT", or "BUS".

Frequency-Shift Keying (FSK) Commands

See also "FSK Modulation," starting on page 90 in chapter 3.

FSK Overview

The following is an overview of the steps required to generate an FSK waveform. The commands used for FSK are listed on the next page.

1 Set up the carrier waveform.

Use the `APPLy` command or the equivalent `FUNC:SHAP`, `FREQ`, `VOLT`, and `VOLT:OFFS` commands to select the function, frequency, amplitude, and offset of the carrier waveform. You can select a sine, square, triangle, ramp, or arbitrary waveform for the carrier.

2 Select the FSK "hop" frequency.

Use the `FSK:FREQ` command to set the "hop" frequency.

3 Set the FSK rate.

Use the `FSK:INT:RATE` command to set rate at which the output frequency "shifts" between the carrier frequency and the hop frequency (*internal FSK source only*).

4 Select the FSK source.

Select an internal or external FSK source using the `FSK:SOUR` command.

5 Enable FSK modulation.

After you have set up the other FSK parameters, use the `FSK:STAT ON` command to enable FSK modulation.

FSK Commands

> *Use the* APPLy *command or the equivalent* FUNC:SHAP, FREQ, VOLT, *and* VOLT:OFFS *commands to configure the carrier waveform. Set the carrier frequency between 10 mHz and 15 MHz (100 kHz for triangle and ramp). The default is 1 kHz.*

FSKey:FREQuency {<*frequency***>|MINimum|MAXimum}**
Set the FSK "hop" frequency. Select from 10 mHz to 15 MHz (100 kHz for triangle and ramp). *The default is 100 Hz*. MIN = 10 mHz. MAX = 15 MHz. [*Stored in volatile memory*]

FSKey:FREQuency? [MINimum|MAXimum]
Query the FSK "hop" frequency. Returns a value in hertz.

FSKey:INTernal:RATE {<*rate in Hz***>|MINimum|MAXimum}**
Set the rate at which the output frequency "shifts" between the carrier and hop frequency (FSK:SOUR INT only). Select from 10 mHz to 50 kHz. *The default is 10 Hz.* MIN = 10 mHz. MAX = 50 kHz.
[*Stored in volatile memory*]

The FSK rate is ignored when the external source is selected (FSK:SOUR EXT).

FSKey:INTernal:RATE? [MINimum|MAXimum]
Query the FSK rate. Returns a value in hertz.

FSKey:SOURce {INTernal|EXTernal}
Select an internal or external FSK source. *The default is INT*.
[*Stored in volatile memory*]

- When the *internal* source is selected, the rate at which the output frequency "shifts" between the carrier frequency and hop frequency is determined by the *FSK rate* specified (FSK:INT:RATE).
- When the *external* source is selected, the output frequency is determined by the signal level on the rear-panel *FSK* terminal. When a "low" TTL level is present, the carrier frequency is output. When a "high" TTL level is present, the hop frequency is output.
- The maximum external FSK rate is 1 MHz.
- The **Ext** annunciator turns on when the function generator is waiting for an external trigger signal.

FSKey:SOURce?
Query the internal FSK source. Returns "INT" or "EXT".

FSKey:STATe {OFF|ON}
Disable or enable FSK modulation. To ensure proper operation, you should enable FSK *after* you have set up the other modulation parameters. Only one modulation mode can be enabled at a time. When you enable FSK, the previous modulation mode is turned off.

FSKey:STATe?
Query the state of the FSK mode. Returns "0" (OFF) or "1" (ON).

Frequency Sweep Commands

See also "Frequency Sweep," starting on page 94 in chapter 3.

Sweep Overview

The following is an overview of the steps required to generate a frequency sweep. The commands used for frequency sweep are listed on the next page.

1 Select the waveform shape, amplitude, and offset.

Use the `APPLy` command or the equivalent `FUNC:SHAP`, `VOLT`, and `VOLT:OFFS` commands to select the function, amplitude, and offset of the sweep waveform. You can select a sine, square, triangle, ramp, or arbitrary waveform.

2 Select the start and stop frequencies.

Use the `FREQ:STAR` and `FREQ:STOP` commands to set the start and stop frequencies respectively.

*To sweep **up** in frequency, set the start frequency* **<** *stop frequency.*
*To sweep **down** in frequency, set the start frequency* **>** *stop frequency.*

3 Set the sweep time.

Use the `SWE:TIME` command to set the number of seconds required to sweep from the start frequency to the stop frequency.

4 Select the sweep mode.

Use the `SWE:SPAC` command to select *linear* or *logarithmic* spacing.

5 Select the sweep trigger source.

Use the `TRIG:SOUR` command to select the source from which the sweep will be triggered.

6 Enable sweep modulation.

Use the `SWE:STAT ON` command to enable the sweep mode.

Sweep Commands

> *To sweep **up** in frequency, set the start frequency* **<** *stop frequency.*
> *To sweep **down** in frequency, set the start frequency* **>** *stop frequency.*

FREQuency:STARt {<*frequency***>|MINimum|MAXimum}**
Set the start frequency. Select from 10 mHz to 15 MHz (100 kHz for triangle and ramp). The sweep is phase continuous over the full frequency range. *The default is 100 Hz*. MIN = 10 mHz. MAX = 15 MHz. [*Stored in volatile memory*]

FREQuency:STARt? [MINimum|MAXimum]
Query the start frequency. Returns a value in hertz.

FREQuency:STOP {<*frequency***>|MINimum|MAXimum}**
Set the stop frequency. Select from 10 mHz to 15 MHz (100 kHz for triangle and ramp). The sweep is phase continuous over the full frequency range. *The default is 1 kHz.* MIN = 10 mHz. MAX = 15 MHz. [*Stored in volatile memory*]

FREQuency:STOP? [MINimum|MAXimum]
Query the stop frequency. Returns a value in hertz.

SWEep:SPACing {LINear|LOGarithmic}
Select linear or logarithmic spacing for the sweep. *The default is Linear*. [*Stored in volatile memory*]

SWEep:SPACing?
Query the sweep mode. Returns “LIN” or “LOG”.

SWEep:TIME {<*seconds***>|MINimum|MAXimum}**
Set the number of seconds required to sweep from the start frequency to the stop frequency. Select from 1 ms to 500 seconds. *The default is 1 second*. MIN = 1 ms. MAX = 500 seconds. [*Stored in volatile memory*]

The number of frequency points in the sweep depends on the sweep time you select and is automatically calculated by the function generator.

SWEep:TIME? [MINimum|MAXimum]
Query the sweep time. Returns a value in seconds.

SWEep:STATe {OFF|ON}
Disable or enable the sweep mode. To ensure proper operation, you should enable the sweep mode *after* you have set up the other sweep parameters. Only one modulation mode can be enabled at a time. When you enable the sweep mode, the previous modulation mode is turned off.

SWEep:STATe?
Query the state of the sweep mode. Returns "0" (OFF) or "1" (ON).

TRIGger:SOURce {IMMediate|EXTernal|BUS}
Select the trigger source. In the *triggered sweep* mode, the function generator outputs a single sweep each time a trigger is received. After one sweep from the start frequency to the stop frequency, the function generators waits for the next trigger while outputting the start frequency. *The default is IMM.* [*Stored in volatile memory*]

- When the *Immediate* (internal) source is selected, the function generator outputs a continuous sweep at a rate determined by the sweep rate specified (`SWE:TIME`). The `APPLy` command automatically sets the trigger source to `IMMediate`.
- When the *External* source is selected, the function generator will accept a hardware trigger applied to the rear-panel *Ext Trig* terminal. The function generator outputs one sweep each time each time *Ext Trig* receives the rising edge of a TTL pulse. The **Trig** annunciator turns on when the function generator is waiting for an external trigger.
- When the *Bus* (software) source is selected, the function generator outputs one sweep each time a bus trigger command is received. To trigger the function generator from the remote interface (GPIB or RS-232), send the `*TRG` (trigger) command.
- To ensure synchronization when the *Bus* source is selected, send the `*WAI` (wait) command. When the `*WAI` command is executed, the function generator waits for all pending operations to complete before executing any additional commands. For example, the following command string guarantees that the first trigger is accepted and executed before the second trigger is recognized.

```
"TRIG:SOUR BUS;*TRG;*WAI;*TRG;*WAI"
```

- You can use the `*OPC?` (operation complete query) command or the `*OPC` (operation complete) command to signal when the sweep is complete. The `*OPC?` command returns “1” to the output buffer when the sweep is complete. The `*OPC` command sets the “operation complete” bit (bit 0) in the Standard Event register when the burst is complete.

`TRIGger:SOURce?`
Query the present trigger source. Returns “IMM”, “EXT”, or “BUS”.

Arbitrary Waveform Commands

See also "Arbitrary Waveforms" starting on page 103 in chapter 3.

Arbitrary Waveform Overview

The following is an overview of the steps required to download and output an arbitrary waveform over the remote interface. The commands used for arbitrary waveforms are listed on page 176. Refer to chapter 7, "Tutorial", for more information on the internal operation of downloading and outputting an arbitrary waveform.

> *Chapter 6, "Application Programs", contains several example programs which show the use of arbitrary waveforms. You may find it useful to refer to the programs after reading the following section in this chapter.*

1 Select the waveform frequency, amplitude, and offset.

Use the `APPLy` command or the equivalent `FREQ`, `VOLT`, and `VOLT:OFFS` commands to select the frequency, amplitude, and offset of the arbitrary waveform.

2 Download the data points into volatile memory.

You can download between 8 and 16,000 points per waveform. The waveform can be downloaded as floating-point values or binary integer values. Use the `DATA VOLATILE` command to download floating-point values between -1 and +1. Use the `DATA:DAC VOLATILE` command to download binary integer values between -2047 and +2047.

To ensure that binary data is downloaded properly, you must select the order in which the bytes are downloaded using the `FORM:BORD` *command.*

3 Copy the arbitrary waveform to non-volatile memory.

You can output the arbitrary waveform directly from volatile memory (as described in step 2) or you can copy the waveform to non-volatile memory. Use the `DATA:COPY` command to copy the waveform to non-volatile memory.

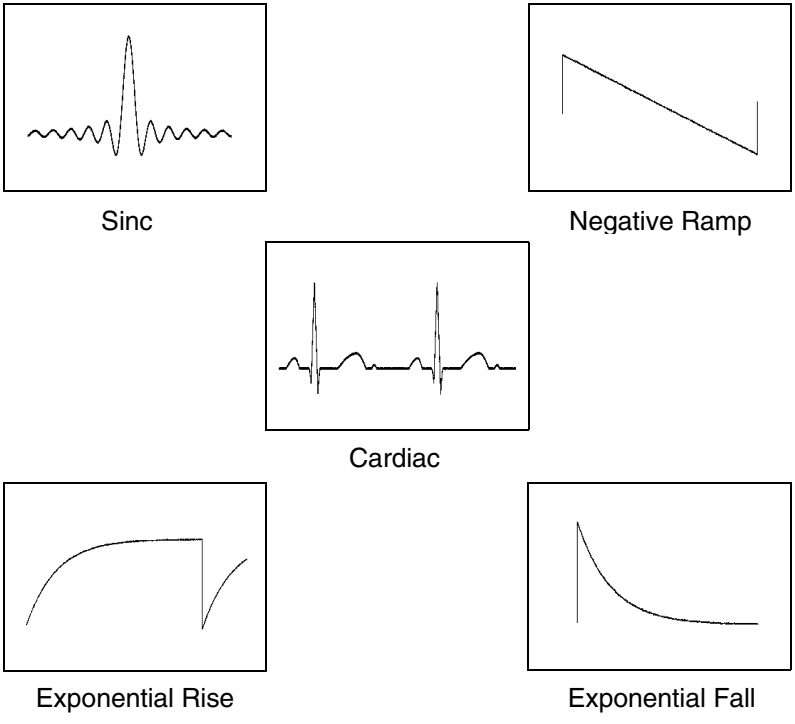
4 Select the arbitrary waveform to output.

You can select one of the five built-in arbitrary waveforms, one of four user-defined waveforms, or the waveform currently downloaded to volatile memory. Use the `FUNC:USER` command to select the waveform.

5 Output the currently selected arbitrary waveform.

Use the `FUNC:SHAP USER` command to output the waveform previously selected with the `FUNC:USER` command.

The five built-in arbitrary waveforms are shown below.



Arbitrary Waveform Commands

FUNCtion:USER {<*arb name***>|VOLATILE}**
Select one of the five built-in arbitrary waveforms, one of four user-defined waveforms, or the waveform currently downloaded to VOLATILE memory.

- The names of the five built-in arbitrary waveforms are: "SINC", "NEG_RAMP", "EXP_RISE", "EXP_FALL", and "CARDIAC".
- To select the waveform currently stored in volatile memory, specify the VOLATILE parameter. The keyword "VOLATILE" does not have a short form. The correct syntax is: `"FUNC:USER VOLATILE"`
- The `FUNC:USER` command *does not* output the selected waveform. Use the `FUNC:SHAP USER` command to output the waveform.
- If you select an arbitrary waveform name that is not currently downloaded, a *+785, "Specified arb waveform does not exist"* error is generated.
- The arb name may contain up to 8 characters. The first character *must* be a letter (A-Z), but the remaining characters can be numbers (0-9) or the underscore character (" _ "). Blank spaces are not allowed. If you specify a name with more than 8 characters, a *+783, "Arb waveform name too long"* error is generated.
- The function generator *does not* distinguish between upper- and lower-case letters for the arbitrary waveform name. Therefore, **ARB_1** and **arb_1** are the same name. All characters are converted to upper case.
- Use the `DATA:CAT?` command to list the names of the five built-in waveforms (non-volatile), "VOLATILE" if a waveform is currently downloaded to volatile memory, and the names of any user-defined waveforms (non-volatile).
- The following statement shows how to use the `FUNC:USER` command: `"FUNC:USER NEG_RAMP"`

FUNCtion:USER?
Query the arbitrary waveform currently selected. Returns "SINC", "NEG_RAMP", "EXP_RISE", "EXP_FALL", "CARDIAC", "VOLATILE", or the name of any user-defined waveforms in non-volatile memory.

FUNCtion:SHAPe USER
Select the function and output the selected arbitrary waveform. When executed, this command outputs the arbitrary waveform currently selected by the FUNC:USER command. The selected waveform is output using the previously selected frequency, amplitude, and offset settings.

- Use the FREQ, VOLT, and VOLT:OFFS commands to select the frequency, amplitude, and offset of the waveform. Or, use the APPLy command to select the function, frequency, amplitude, and offset with a single command.
- As shown in the following table, you can use the arbitrary waveform function with all of the modulation modes. Each "X" indicates a valid combination.

	Sine	Square	Triangle	Ramp	Noise	Arb
AM Carrier	X	X	X	X		X
AM Modulating Wave	X	X	X	X	X	X
FM Carrier	X	X	X	X		X
FM Modulating Wave	X	X	X	X	X	X
FSK Modulation	X	X	X	X		X
Burst Modulation	X	X	X	X		X
Frequency Sweep	X	X	X	X		X

- For arbitrary waveforms, the maximum amplitude will be limited if the data points do not span the full range of the output DAC (Digital-to-Analog Converter). For example, the built-in "SINC" waveform does not use the full range of values between ±1 and therefore its maximum amplitude is 6.084 Vpp (into 50 ohms).
- For arbitrary waveforms, the **Offset** annunciator will turn on if the waveform data has an inherent offset present (if the average is not equal to zero). The function generator calculates the average of the data points and compares the average to zero volts. If the average is not within two DAC (Digital-to-Analog Converter) counts of zero volts, the **Offset** annunciator turns on.

FUNCtion:SHAPe?
Query the output function. Returns "SIN", "SQU", "TRI", "RAMP", "NOIS", "DC", or "USER".

DATA VOLATILE, <*value*>, <*value*>, . . .
Download *floating-point* values between *-1* and *+1* into volatile memory. You can download between 8 and 16,000 points per waveform.

- The values -1 and +1 correspond to the peak values of the waveform. For example, if you set the amplitude to 10 Vpp, "-1" corresponds to -5 volts and "+1" corresponds to +5 volts.
- The maximum amplitude will be limited if the data points do not span the full range of the output DAC (Digital-to-Analog Converter). For example, the built-in "SINC" waveform does not use the full range of values between ±1 and therefore its maximum amplitude is 6.084 Vpp (into 50 ohms).
- Downloading floating-point values (using `DATA VOLATILE`) is slower than downloading binary values (using `DATA:DAC VOLATILE`) but is more convenient when using trigonometric functions which return values between -1 and +1.
- The `DATA VOLATILE` command overwrites the previous waveform in VOLATILE memory (no error is generated). Use the `DATA:COPY` command to copy the waveform to non-volatile memory.
- Up to four user-defined waveforms can be stored in non-volatile memory. Use the `DATA:DEL` command to delete the waveform in VOLATILE memory or any of the four user-defined waveforms in non-volatile memory. Use the `DATA:CAT?` command to list all waveforms currently stored in volatile and non-volatile memory (and the built-in waveforms).
- After downloading the waveform data to memory, use the `FUNC:USER` command to choose the active waveform and the `FUNC:SHAP USER` command to output it.
- The following statement shows how to use the `DATA VOLATILE` command to download nine points to volatile memory:

```
"DATA VOLATILE, 1,.75,.5,.25,0,-.25,-.5,-.75,-1"
```

DATA:DAC VOLATILE, {<*binary block***>|<***value***>, <***value***>, . . . }**
Download *binary integer* values between *-2047* and *+2047* into volatile memory. You can download between 8 and 16,000 points per waveform in IEEE-488.2 binary block format or as a list of values. The binary range of values corresponds to the values available using internal 12-bit DAC (Digital-to-Analog Converter) codes.

- The values -2047 and +2047 correspond to the peak values of the waveform. For example, if you have set the output amplitude to 10 Vpp, "-2047" corresponds to -5 volts and "+2047" corresponds to +5 volts.
- The maximum amplitude will be limited if the data points do not span the full range of the output DAC (Digital-to-Analog Converter). For example, the built-in "SINC" waveform does not use the full range of values between ±1 and therefore its maximum amplitude is 6.084 Vpp (into 50 ohms).
- The `DATA:DAC VOLATILE` command overwrites the previous waveform in VOLATILE memory (no error is generated). Use the `DATA:COPY` command to copy the waveform to non-volatile memory.
- Up to four user-defined waveforms can be stored in non-volatile memory. Use the `DATA:DEL` command to delete the waveform in VOLATILE memory or any of the four user-defined waveforms in non-volatile memory. Use the `DATA:CAT?` command to list all waveforms currently stored in volatile and non-volatile memory (and the built-in waveforms).
- After downloading the waveform data to memory, use the `FUNC:USER` command to choose the active waveform and the `FUNC:SHAP USER` command to output it.

DATA:DAC VOLATILE *(continued)*

- To download binary data over the RS-232 interface, you must select *8 data bits* with *no parity*. *See "RS-232 Interface Configuration" on page 195 for more information.*
- The following statement shows how to use the `DATA:DAC VOLATILE` command to download eight integer points using the binary block format (*see also "Using the IEEE-488.2 Binary Block Format" below*):

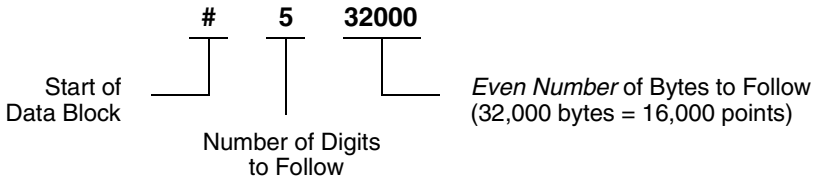
  `"DATA:DAC VOLATILE, #216` *Binary Data* `"`

- The following statement shows how to use the `DATA:DAC VOLATILE` command to download eight integer points:

```
"DATA:DAC VOLATILE, 2047,1536,1024,512,0,-512,-1536,-2047"
```

Using the IEEE-488.2 Binary Block Format

In the binary block format, a *block header* precedes the waveform data. The block header has the following format:



The function generator represents binary data as 12-bit integers, which is sent as two bytes. Therefore, **the total number of bytes is always twice the number of data points in the waveform** (and must always be an *even number*). For example, 32,000 bytes are required to download a waveform with 16,000 points.

Use the FORM:BORD command to select the byte order for binary transfers in block mode. If you specify FORM:BORD NORM (default), the most-significant byte (MSB) of each data point is sent first. If you specify FORM:BORD SWAP, the least-significant byte (LSB) of each data point is sent first. Most PCs use the "swapped" byte order.

DATA:ATTRibute:AVERage? [<*arb name*>]
Query the arithmetic *average* of all data points for the specified arbitrary waveform. The default *arb name* is the arbitrary waveform currently active (selected with FUNC:USER command).

- If you query a waveform that is not currently stored in memory, a +*785, "Specified arb waveform does not exist"* error is generated.

DATA:ATTRibute:CFACtor? [<*arb name*>]
Query the *crest factor* of all data points for the specified arbitrary waveform. Crest factor is the ratio of the peak value to the RMS value of the waveform. The default *arb name* is the arbitrary waveform currently active (selected with FUNC:USER command).

- If you query a waveform that is not currently stored in memory, a +*785, "Specified arb waveform does not exist"* error is generated.

DATA:ATTRibute:POINts? [<*arb name*>]
Query the number of points in the specified arbitrary waveform. Returns a value between 8 and 16,000 points. The default *arb name* is the arbitrary waveform currently active (selected with FUNC:USER command).

- If you query a waveform that is not currently stored in memory, a +*785, "Specified arb waveform does not exist"* error is generated.

DATA:ATTRibute:PTPeak? [<*arb name*>]
Query the *peak-to-peak* value of all data points for the specified arbitrary waveform. The default *arb name* is the arbitrary waveform currently active (selected with FUNC:USER command).

- This command returns a value between "0" and "+1.0", with "+1.0" indicating full amplitude available (100%).
- The maximum peak-to-peak amplitude will be limited if the data points do not span the full range of the output DAC (Digital-to-Analog Converter). For example, the built-in "SINC" waveform does not use the full range of values between ±1 and therefore its maximum peak-to-peak amplitude is 6.084 Vpp (into 50 ohms).
- If you query a waveform that is not currently stored in memory, a +*785, "Specified arb waveform does not exist"* error is generated.

DATA:CATalog?
List the names of *all* waveforms currently downloaded to memory. Returns the names of the five built-in waveforms (non-volatile memory), "VOLATILE" if a waveform is currently downloaded to volatile memory, and all user-defined waveforms downloaded to non-volatile memory.

- A series of quoted strings separated with commas is returned:

```
"SINC","NEG_RAMP","EXP_RISE","EXP_FALL",
"CARDIAC","VOLATILE","ARB_1","ARB_2"
```

- Use the DATA:DEL command to delete the waveform in VOLATILE memory or any of the user-defined waveforms in non-volatile memory.

DATA:COPY <*destination arb name***> [,VOLATILE]**
Copy the waveform from volatile memory to the specified name in non-volatile memory. The source for the copy is always "VOLATILE" (you cannot copy from any other name) and you cannot *copy to* VOLATILE. The VOLATILE parameter is optional and can be omitted.

- The arb name may contain up to 8 characters. The first character *must* be a letter (A-Z), but the remaining characters can be numbers (0-9) or the underscore character (" _ "). Blank spaces are not allowed. If you specify a name with more than 8 characters, a *+783, "Arb waveform name too long"* error is generated.
- The following built-in waveform names are reserved and cannot be used with the DATA:COPY command: SINC, NEG_RAMP, EXP_RISE, EXP_FALL, and CARDIAC. If you specify one of the built-in waveforms, a *+782, "Cannot overwrite a built-in arb waveform"* error is generated.
- The function generator *does not* distinguish between upper- and lower-case letters. Therefore, **ARB_1** and **arb_1** are the same name. All characters are converted to upper case.

- If you copy to a waveform name that already exists, the previous waveform is overwritten (no error is generated). However, you cannot overwrite any of the five built-in waveforms.
- Up to four user-defined waveforms can be stored in non-volatile memory. If memory is full and you try to copy a new waveform to non-volatile memory, a *+781, "Not enough memory"* error is generated. Use the `DATA:DEL` command to delete the waveform in VOLATILE memory or any of the four user-defined waveforms in non-volatile memory. Use the `DATA:CAT?` command to list all waveforms currently stored in volatile and non-volatile memory.
- The following statement shows how to use the `DATA:COPY` command:

```
"DATA:COPY ARB_1, VOLATILE"
```

`DATA:DELete` **<***arb name***>**
Delete the specified arbitrary waveform from memory. You can delete the waveform in volatile memory or any of the four user-defined waveforms in non-volatile memory.

- You cannot delete the arbitrary waveform that is currently being output. If you attempt to delete the waveform being output, a *+787, "Cannot delete an active arb waveform"* error is generated.
- You cannot delete any of the five built-in arbitrary waveforms. If you attempt to delete one of the built-in waveforms, a *+786, "Cannot delete a built-in arb waveform"* error is generated.
- Use the `DATA:DEL:ALL` command to delete the waveform in VOLATILE memory and all user-defined non-volatile waveforms *all at once*. If one of the waveforms is currently being output, a *+787, "Cannot delete an active arb waveform"* error is generated. The active waveform *is not* deleted but all of the other waveforms are deleted.

DATA:DELete:ALL
Delete all user-defined arbitrary waveforms from memory. This command deletes the waveform in VOLATILE memory and all user-defined waveforms in non-volatile memory. The five built-in waveforms in non-volatile memory *are not* deleted.

- The colon before the ALL parameter is required.

  ```
  DATA:DELete:ALL
  ```

  If you insert a space instead of a colon, the function generator will attempt to delete an arbitrary waveform with the name "ALL". If no such waveform is stored in memory, a *+785, "Specified arb waveform does not exist"* error is generated.

- You cannot delete the arbitrary waveform that is currently being output. If you attempt to delete the waveform being output, a *+787, "Cannot delete an active arb waveform"* error is generated.
- You cannot delete any of the five built-in arbitrary waveforms. If you attempt to delete one of the built-in waveforms, a *+786, "Cannot delete a built-in arb waveform"* error is generated.
- Use the DATA:DEL <*arb name*> command to delete stored waveforms *one at a time*.

DATA:NVOLatile:CATalog?
List the names of all user-defined arbitrary waveforms currently downloaded to non-volatile memory. Returns the names of up to four waveforms in non-volatile memory.

- A series of quoted strings separated with commas is returned (see example below). If no user-defined waveforms are currently downloaded to non-volatile memory, the command returns a null string ("").

  ```
  "ARB_1","ARB_2","ARB_3","ARB_4"
  ```

- Use the DATA:DEL command to delete the waveform in VOLATILE memory or any of the user-defined waveforms in non-volatile memory.

DATA:NVOLatile:FREE?
Query the number of non-volatile *memory slots* available to store user-defined waveforms. Non-volatile waveform memory is divided into four 16k-point slots. This command returns the number of memory slots available to store user-defined waveforms: "0" (memory is full), "1", "2", "3", or "4".

FORMat:BORDer {NORMal|SWAPped}
Used only for binary block transfers. Select the byte order for binary transfers in the block mode using the DATA:DAC command. *The default is NORM*.

- In *NORM* byte order (default), the most-significant byte (MSB) of each data point is sent first.
- In *SWAP* byte order, the least-significant byte (LSB) of each data point is sent first. Most PCs use the "swapped" byte order.
- The function generator represents binary data as 12-bit integers, which are sent as two bytes. Each data point in the waveform requires 12 bits, which must be transferred as 16 bits (two bytes) on the function generator's 8-bit interface.

FORMat:BORDer?
Query the byte order configuration. Returns "NORM" or "SWAP".

SCPI

First-time SCPI users, see page 211.

Triggering

See also "Triggering," starting on page 98 in chapter 3.

Applies only to burst modulation and frequency sweep. You can issue triggers for bursts and sweeps using an *immediate* trigger, an *external* trigger, or a *bus* trigger.

`TRIGger:SOURce {IMMediate|EXTernal|BUS}`
Select the source from which the function generator will accept a trigger. The function generator will accept an immediate internal trigger, a hardware trigger from the rear-panel *Ext Trig* terminal, or a software (bus) trigger. *The default is Immediate*.
[*Stored in volatile memory*]

- When the *Immediate* (internal) source is selected, the function generator outputs continuously when the burst mode or sweep mode is enabled. The `APPLy` command automatically sets the trigger source to `IMMediate`.
- When the *External* source is selected, the function generator will accept a hardware trigger applied to the rear-panel *Ext Trig* terminal. The function generator outputs one burst or initiates one sweep each time each time *Ext Trig* receives the rising edge of a TTL pulse. The **Trig** annunciator turns on when the function generator is waiting for an external trigger.
- When the *Bus* (software) source is selected, the function generator outputs one burst or initiates one sweep each time a bus trigger command is received. To trigger the function generator from the remote interface (GPIB or RS-232), send the `*TRG` (trigger) command. *You can also trigger the function generator from the GPIB interface by sending the IEEE-488 Group Execute Trigger (GET) message* (e.g., `TRIGGER 710`).

- To ensure synchronization when the *Bus* source is selected, send the `*WAI` (wait) command. When the `*WAI` command is executed, the function generator waits for all pending operations to complete before executing any additional commands. For example, the following command string guarantees that the first trigger is accepted and executed before the second trigger is recognized.

```
"TRIG:SOUR BUS;*TRG;*WAI;*TRG;*WAI"
```

- You can use the `*OPC?` (operation complete query) command or the `*OPC` (operation complete) command to signal when the burst or sweep is complete. The `*OPC?` command returns "1" to the output buffer when the burst or sweep is complete. The `*OPC` command sets the "operation complete" bit (bit 0) in the Standard Event register when the burst or sweep is complete.

TRIGger:SOURce?
Query the present trigger source. Returns "IMM", "EXT", or "BUS".

TRIGger:SLOPe {POSitive|NEGative}
Select the polarity of the *Ext Trig* terminal. If you select "POS", the polarity of the *Ext Trig* terminal is the same as described on page 102. If you select "NEG", the polarity is the opposite of what is described. *The default is Positive.* [*Stored in volatile memory*]

TRIGger:SLOPe?
Query the present polarity of the *Ext Trig* terminal. Returns "POS" or "NEG".

***TRG**
Trigger the function generator from the remote interface. For RS-232 operation, make sure the interface is in the remote mode by sending the `SYST:REM` command.

System-Related Commands

See also "System-Related Operations," starting on page 109 in chapter 3.

DISPlay {OFF|ON}
Turn the front-panel display off or on. When the display is turned off, output parameters are not sent to the display and all annunciators except **ERROR** and **Shift** are disabled. Front-panel operation is otherwise unaffected by turning off the display. [*Stored in volatile memory*]

- Sending a message to the display from the remote interface overrides the display state; this means that you can display a message even if the display is turned off.
- The display state is automatically turned on when you return to the local (front panel) state. Press the Shift key or execute `LOCAL 710` from the remote interface to return to the local state.

DISPlay?
Query the front-panel display setting. Returns "0" (OFF) or "1" (ON).

DISPlay:TEXT <*quoted string***>**
Display a message containing up to 11 characters on the front panel; any additional characters are truncated. Commas, periods, and semicolons share a display space with the preceding character, and *are not* considered individual characters. When a message is displayed, information relating to the output waveform (such as frequency and amplitude) is not sent to the display. [*Stored in volatile memory*]

The `DISP:TEXT` *command is allowed only while the function generator is in the remote mode. An error is generated if you attempt to execute the command while in the local mode.*

The following command string shows how to display a message on the front panel.

```
"DISP:TEXT 'HELLO'"
```

DISPlay:TEXT?
Query the message sent to the front panel and return a quoted string. For example, the query returns a string such as `"HELLO"`.

DISPlay:TEXT:CLEar
Clear the message displayed on the front panel.

SYSTem:BEEPer
Issue a single beep immediately.

SYSTem:ERRor?
Read one error from the *error queue.* When the front-panel **ERROR** annunciator turns on, one or more command syntax or hardware errors have been detected. A record of up to 20 errors can be stored in the function generator's error queue. *See chapter 5, "Error Messages," for a complete listing of the errors.*

- Errors are retrieved in first-in-first-out (FIFO) order. The first error returned is the first error that was stored. When you have read all errors from the queue, the **ERROR** annunciator turns off. The function generator beeps once each time an error is generated.
- If more than 20 errors have occurred, the last error stored in the queue (the most recent error) is replaced with *-350, "Too many errors"*. No additional errors are stored until you remove errors from the queue. If no errors have occurred when you read the error queue, the function generator responds with *+0, "No error"*.
- The error queue is cleared when power has been off or after a `*CLS` (clear status) command has been executed. The `*RST` (reset) command *does not* clear the error queue.
- Errors have the following format (the error string may contain up to *80 characters*):

```
-113,"Undefined header"
```

SYSTem:VERSion?
Query the function generator to determine the present SCPI version. Returns a string in the form "YYYY.V" where the "Y's" represent the year of the version, and the "V" represents a version number for that year (for example, 1993.0).

***IDN?**
Read the function generator's identification string. The function generator returns four fields separated by commas. The *first* field is the manufacturer's name, the *second* field is the model number, the *third* field is not used (always "0"), and the *fourth* field is a revision code which contains three numbers. The first number is the firmware revision number for the *main generator processor*; the second is for the *input/output processor*; and the third is for the *front-panel processor*.

- The command returns a string with the following format (be sure to dimension a string variable with at least *40 characters*):

```
HEWLETT-PACKARD,33120A,0,X.X-X.X-X.X
```

***RST**
Reset the function generator to its default state (*see "Power-On and Reset State" on page 123*) independent of the POWER ON command setting in the front-panel SYS MENU.

In the sweep mode, the `*RST` *command* ***does not*** *set the start and stop frequencies to their default values. Instead, the reset operation sets the start frequency to 10 mHz (minimum) and the stop frequency to 15 MHz (maximum).*

***TST?**
Perform a complete self-test of the function generator. Returns "+0" if the self-test passes or "+1" if it fails. If the self-test fails, an error message is also generated with additional information on why the test failed.

***SAV {0|1|2|3}**
Store up to four different instrument configurations.
[*Stored in non-volatile memory*]

- Four memory locations (numbered 0, 1, 2, and 3) are available to store instrument configurations. The state storage feature "remembers" the function (including arbitrary waveforms), frequency, amplitude, dc offset, duty cycle, as well as any modulation parameters. To recall a stored state, you must use the same memory location used previously to store the state.
- From the remote interface only, you can use memory location "0" to store a fourth instrument configuration (you cannot store to this memory location from the front panel). However, keep in mind that memory location "0" is automatically overwritten when the power is turned off.
- Any arbitrary waveforms downloaded to "VOLATILE" memory *are not* remembered. However, if an arbitrary waveform is being output from *non-volatile* memory when the state is stored, the waveform data is stored. The stored waveform is output when the instrument state is recalled.
- If you delete an arbitrary waveform after storing the state, the waveform data is lost and the function generator will not output the waveform when the state is recalled. The "SINC" waveform is output in place of the deleted waveform.
- When power is turned off, the function generator automatically stores its state in memory location "0". You can configure the function generator to automatically recall the power-down state when power is restored. The recall mode is *disabled* when the function generator is shipped from the factory.

  Select the POWER ON LAST STATE command from the SYS MENU to enable the power-down recall mode. Select POWER ON DEFAULT to disable the power-down recall mode. *See "Power-Down Recall Mode" on page 109 for more information.*

***RCL {0|1|2|3}**
Recall a previously stored state. To recall a stored state, you must use the same memory location used previously to store the state.

- You cannot recall the instrument state from a memory location that was not previously specified as a storage location. For example, an error is generated if you attempt to recall from memory location "2" but have never stored to that location. *From the remote interface, a +810, "State has not been stored" error is generated if nothing is stored in the specified memory location.*

MEMory:STATe:DELete {0|1|2|3}
Delete a previously stored state and clear the memory location.
If nothing is stored in the specified memory location, a *+810, "State has not been stored"* error is generated.

***OPC**
Set the "operation complete" bit (bit 0) in the Standard Event register after the previous commands have been executed. *Used only in the triggered burst mode and triggered sweep mode.*

***OPC?**
Return "1" to the output buffer after the previous commands have been executed. *Used only in the triggered burst mode and triggered sweep mode.*

***WAI**
Wait for all pending operations to complete before executing any additional commands over the interface. *Used only in the triggered burst mode and triggered sweep mode.*

Calibration Commands

See "Calibration Overview" starting on page 118 for an overview of the calibration features of the function generator. For a more detailed discussion of the calibration procedures, see chapter 4 in the Service Guide.

CALibration?
Perform a calibration using the specified calibration value (CAL:VAL command). Before you can calibrate the function generator, you must unsecure it by entering the correct security code.

CALibration:COUNt?
Query the function generator to determine the number of times it has been calibrated. Your function generator was calibrated before it left the factory. When you receive your function generator, read the count to determine its initial value. [*Stored in non-volatile memory*]

- The calibration count increments up to a maximum of 32,767 after which it wraps-around to 0. Since the value increments by one for each calibration point, a complete calibration will increase the value by many counts.

CALibration:SECure:CODE <*new code***>**
Enter a new security code. To change the security code, you must first unsecure the function generator using the old security code, and then enter a new code. The calibration code may contain up to 12 characters. [*Stored in non-volatile memory*]

CALibration:SECure:STATe {OFF|ON},<*code***>**
Unsecure or secure the function generator for calibration. The calibration code may contain up to 12 characters. [*Stored in non-volatile memory*]

CALibration:SECure:STATe?
Query the secured state of the function generator. Returns "0" (OFF) or "1" (ON).

CALibration:SETup <0|1|2|3| ... |84>
Configure the function generator's internal state for each of the calibration steps to be performed.

CALibration:SETup?
Query the calibration setup number. Returns a value between 0 and 84.

CALibration:STRing <*quoted string***>**
Record calibration information about your function generator. For example, you can store such information as the last calibration date, the next calibration due date, the instrument serial number, or even the name and phone number of the person to contact for a new calibration. [*Stored in non-volatile memory*]

- You can record information in the calibration message only from the remote interface. However, you can read the message from either the front-panel menu or the remote interface.
- The calibration message may contain up to 40 characters. However, the function generator can display only 11 characters of the message on the front panel (additional characters are truncated).

CALibration:STRing?
Query the calibration message and return a quoted string.

CALibration:VALue <*value***>**
Specify the value of the known calibration signal used by the calibration procedure. Use the `CAL:SET` command to configure the function generator's internal state for each of the calibration steps to be performed.

CALibration:VALue?
Query the present calibration value.

RS-232 Interface Configuration

See also "Remote Interface Configuration," on page 114 in chapter 3.

You connect the function generator to the RS-232 interface using the 9-pin (DB-9) serial connector on the rear panel. The function generator is configured as a DTE (*Data Terminal Equipment*) device. For all communications over the RS-232 interface, the function generator uses two handshake lines: DTR (*Data Terminal Ready*) on pin 4 and DSR (*Data Set Ready*) on pin 6.

The following sections contain information to help you use the function generator over the RS-232 interface. The programming commands for RS-232 are listed on page 200.

RS-232 Configuration Overview

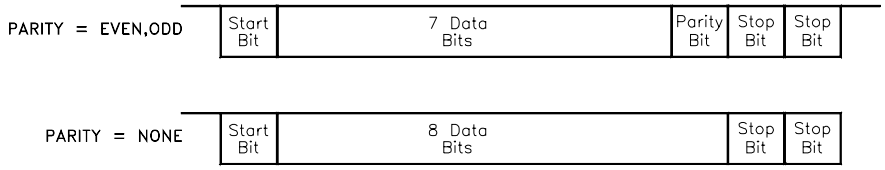
Configure the RS-232 interface using the parameters shown below. Use the front-panel I/O MENU to select the baud rate, parity, and number of data bits (*see also pages 219 and 220 for more information*).

- Baud Rate: 300, 600, 1200, 2400, 4800, or **9600 baud** (*factory setting*)
- Parity and Data Bits: **None / 8 data bits** (*factory setting)*
  Even / 7 data bits, or
  Odd / 7 data bits
- Number of Start Bits: **1 bit** (*fixed*)
- Number of Stop Bits: **2 bits** (*fixed*)

> *To download binary data for arbitrary waveforms over the RS-232 interface, you must select **8 data bits** with **no parity**.*

RS-232 Data Frame Format

A character *frame* consists of all the transmitted bits that make up a single character. The frame is defined as the characters from the *start bit* to the last *stop bit*, inclusively. Within the frame, you can select the baud rate, number of data bits, and parity type. The function generator uses the following frame formats for seven and eight data bits.



Connection to a Computer or Terminal

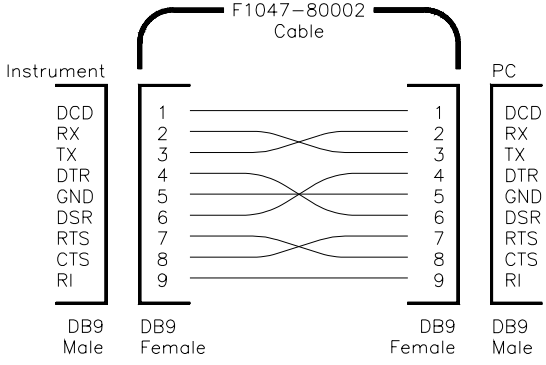
To connect the function generator to a computer or terminal, you must have the proper interface cable. Most computers and terminals are DTE (*Data Terminal Equipment*) devices. Since the function generator is also a DTE device, you must use a DTE-to-DTE interface cable. These cables are also called *null-modem*, *modem-eliminator*, or *crossover* cables.

The interface cable must also have the proper connector on each end and the internal wiring must be correct. Connectors typically have 9 pins (DB-9 connector) or 25 pins (DB-25 connector) with a “male” or “female” pin configuration. A male connector has pins inside the connector shell and a female connector has holes inside the connector shell.

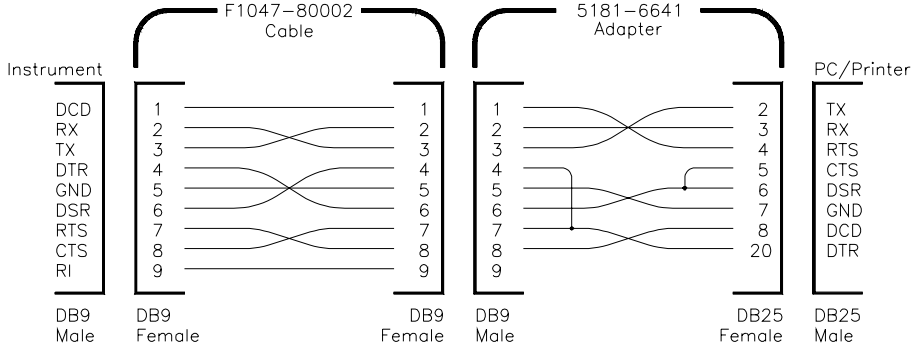
If you cannot find the correct cable for your configuration, you may have to use a *wiring adapter*. If you are using a DTE-to-DTE cable, make sure the adapter is a “straight-through” type. Typical adapters include gender changers, null-modem adapters, and DB-9 to DB-25 adapters.

Refer to the cable and adapter diagrams on the following page to connect the function generator to most computers or terminals. If your configuration is different than those described, order the *34399A Adapter Kit*. This kit contains adapters for connection to other computers, terminals, and modems. Instructions and pin diagrams are included with the adapter kit.

DB-9 Serial Connection If your computer or terminal has a 9-pin serial port with a male connector, use the null-modem cable included with the *34398A Cable Kit*. This cable has a 9-pin female connector on each end. The cable pin diagram is shown below.



DB-25 Serial Connection If your computer or terminal has a 25-pin serial port with a male connector, use the null-modem cable and 25-pin adapter included with the *34398A Cable Kit*. The cable and adapter pin diagram is shown below.



DTR / DSR Handshake Protocol

The function generator is configured as a DTE (*Data Terminal Equipment*) device and uses the DTR (*Data Terminal Ready*) and DSR (*Data Set Ready*) lines of the RS-232 interface to handshake. The function generator uses the DTR line to send a hold-off signal. The DTR line must be TRUE before the function generator will accept data from the interface. When the function generator sets the DTR line FALSE, the data must cease within 10 characters.

To disable the DTR/DSR handshake, *do not* connect the DTR line and tie the DSR line to logic TRUE. If you disable the DTR/DSR handshake, also select a slower baud rate (300, 600, or 1200 baud) to ensure that the data is transmitted correctly.

The function generator sets the DTR line FALSE in the following cases:

1 When the function generator's input buffer is full (when approximately 100 characters have been received), it sets the DTR line FALSE (pin 4 on the RS-232 connector). When enough characters have been removed to make space in the input buffer, the function generator sets the DTR line TRUE, unless the second case (see below) prevents this.

2 When the function generator wants to "talk" over the interface (which means that it has processed a query) and has received a <*new line*> message terminator, it will set the DTR line FALSE. This implies that once a query has been sent to the function generator, the controller should read the response before attempting to send more data. It also means that a <*new line*> must terminate the command string. After the response has been output, the function generator sets the DTR line TRUE again, unless the first case (see above) prevents this.

The function generator monitors the DSR line to determine when the controller is ready to accept data over the interface. The function generator monitors the DSR line (pin 6 on the RS-232 connector) before each character is sent. The output is suspended if the DSR line is FALSE. When the DSR line goes TRUE, transmission will resume.

The function generator holds the DTR line FALSE while output is suspended. A form of interface *deadlock* exists until the controller asserts the DSR line TRUE to allow the function generator to complete the transmission. You can break the interface deadlock by sending the <*Ctrl-C*> character, which clears the operation in progress and discards pending output (this is equivalent to the IEEE-488 device clear action).

For the <Ctrl-C> character to be recognized reliably by the function generator while it holds DTR FALSE, the controller must first set DSR FALSE. When downloading binary data for an arbitrary waveform, press the front-panel LOCAL key first and then send <Ctrl-C>.

RS-232 Troubleshooting

Here are a few things to check if you are having problems communicating over the RS-232 interface. If you need additional help, refer to the documentation that came with your computer.

- Verify that the function generator and your computer are configured for the same baud rate, parity, and number of data bits. Make sure that your computer is set up for *1 start bit* and *2 stop bits* (these values are fixed on the function generator).
- Make sure to execute the `SYSTEM:REMOTE` command to place the function generator in the REMOTE mode.
- Verify that you have connected the correct interface cable and adapters. Even if the cable has the proper connectors for your system, the internal wiring may not be correct. The *34398A Cable Kit* can be used to connect the function generator to most computers or terminals.
- Verify that you have connected the interface cable to the correct serial port on your computer (COM1, COM2, etc).

RS-232 Interface Commands

Use the front-panel I/O MENU to select the baud rate, parity, and number of data bits (see pages 219 and 220 for more information).

SYSTem:LOCal
Place the function generator in the *local* mode for RS-232 operation. All keys on the front panel are fully functional.

SYSTem:REMote
Place the function generator in the *remote* mode for RS-232 operation. All keys on the front panel, except the LOCAL key, are disabled.

> *It is very important that you send the* **SYSTEM:REMOTE** *command to place the function generator in the remote mode. Sending or receiving data over the RS-232 interface when not configured for remote operation can cause unpredictable results.*

SYSTem:RWLock
Place the function generator in the *remote* mode for RS-232 operation. This command is the same as the SYSTEM:REMOTE command except that *all keys* on the front panel are disabled, including the LOCAL key.

Ctrl-C
Clear the operation in progress over the RS-232 interface and discard any pending output data. *This is equivalent to the IEEE-488 device clear action over the GPIB interface.*

The SCPI Status Registers

The function generator uses the *Status Byte* register group and the *Standard Event* register group to record various instrument conditions. A diagram of the SCPI status system is shown on the next page.

> *An example program is included in chapter 6, "Application Programs," which shows the use of the status registers. You may find it useful to refer to the program after reading the following section in this chapter.*

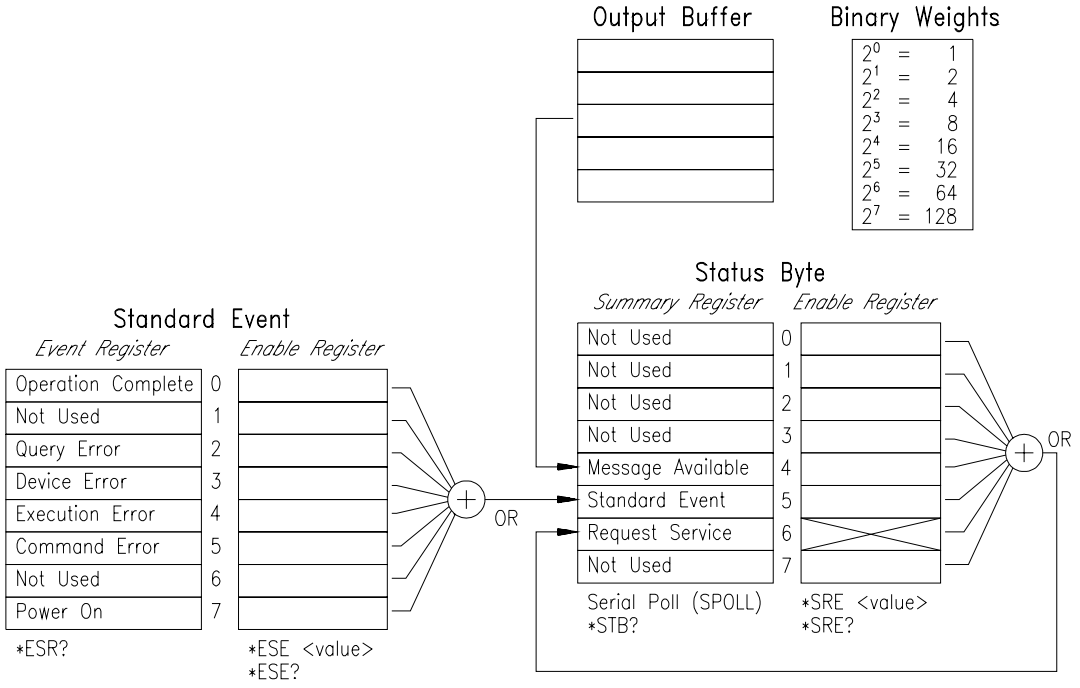
What is an *Event Register*?

An *event register* is a read-only register that reports defined conditions within the function generator. Bits in an event register *are latched*. When an event bit is set, subsequent state changes are ignored. Bits in an event register are automatically cleared by a query of that register or by sending the `*CLS` (clear status) command. The `*RST` (reset) command or device clear will not clear bits in an event register. Querying an event register returns a decimal value which corresponds to the binary-weighted sum of all bits set in the register.

What is an *Enable Register*?

An *enable register* defines which bits in the corresponding event register are logically ORed together to form a single summary bit. You can read from, and write, to an enable register. Querying an enable register *will not* clear it. The `*CLS` (clear status) command does not clear enable registers but it does clear the bits in the event registers. To enable bits in an enable register, you must write a decimal value which corresponds to the binary-weighted sum of the bits you wish to enable in the register.

SCPI Status System



The Status Byte Register

The Status Byte *summary register* reports conditions from the other status registers. Query data that is waiting in the function generator's output buffer is immediately reported through the "message available" bit (bit 4). Bits in the summary register are *not latched*. Clearing an event register will clear the corresponding bits in the Status Byte summary register. Reading all messages in the output buffer, including any pending queries, will clear the message available bit.

Bit Definitions – Status Byte Register

Bit	Decimal Value	Definition
0 Not Used	1	Always set to 0.
1 Not Used	2	Always set to 0.
2 Not Used	4	Always set to 0.
3 Not Used	8	Always set to 0.
4 Message Available	16	Data is available in the output buffer.
5 Standard Event	32	One or more bits are set in the Standard Event register (bits must be "enabled" in enable register).
6 Request Service	64	Requesting service (serial poll).
7 Not Used	128	Always set to 0.

***The Status Byte Summary Register is cleared when*:**

- You execute the `*CLS` (clear status) command.
- Querying the Standard Event register (`*ESR?` command) will clear only bit 4 in the Summary Register.

***The Status Byte Enable Register (request service) is cleared when*:**

- You execute the `*SRE 0` command.
- You turn on the power and have previously configured the function generator using the `*PSC 1` command.
- The enable register *will not* be cleared at power-on if you have previously configured the function generator using `*PSC 0`.

Using Service Request (SRQ) and Serial POLL

You must configure your bus controller to respond to the IEEE-488 service request (SRQ) interrupt to use this capability. Use the Status Byte enable register (`*SRE` command) to select which summary bits will set the low-level IEEE-488 service request signal. When bit 6 (request service) is set in the Status Byte, an IEEE-488 service request interrupt message is automatically sent to the bus controller. The bus controller may then poll the instruments on the bus to identify which one requested service (the instrument with bit 6 set in its Status Byte).

The request service bit is cleared only by reading the Status Byte using an IEEE-488 serial poll or by reading the event register whose summary bit is causing the service request.

To read the Status Byte summary register, send the IEEE-488 serial poll message. Querying the summary register will return a decimal value which corresponds to the binary-weighted sum of the bits set in the register. Serial poll will automatically clear the “request service” bit in the Status Byte summary register. No other bits are affected. Performing a serial poll will not affect instrument throughput.

Caution

The IEEE-488.2 standard does not ensure synchronization between your bus controller program and the instrument. Use the `*OPC?` command to guarantee that commands previously sent to the instrument have completed. Executing a serial poll before a `*RST`, `*CLS`, or other commands have completed can cause previous conditions to be reported.

Using *STB? to Read the Status Byte

The `*STB?` (status byte query) command is similar to a serial poll but it is processed like any other instrument command. The `*STB?` command returns the same result as a serial poll but the "request service" bit (bit 6) *is not* cleared if a serial poll has occurred.

The `*STB?` command is not handled automatically by the IEEE-488 bus interface hardware and will be executed *only* after previous commands have completed. Polling is not possible using the `*STB?` command. Executing the `*STB?` command does not clear the Status Byte summary register.

Using the Message Available Bit (MAV)

You can use the Status Byte "message available" bit (bit 4) to determine when data is available to read into your bus controller. The function generator subsequently clears bit 4 *only* after all messages have been read from the output buffer.

To Interrupt Your Bus Controller Using SRQ

1. Send a device clear message to clear the function generator's output buffer (e.g., `CLEAR 710`).
2. Clear the event registers with the `*CLS` (clear status) command.
3. Set up the enable register masks. Execute the `*ESE` command to set up the Standard Event register and the `*SRE` command for the Status Byte.
4. Send the `*OPC?` (operation complete query) command and enter the result to ensure synchronization.
5. Enable your bus controller's IEEE-488 SRQ interrupt.

To Determine When a Command Sequence is Completed

1. Send a device clear message to clear the function generator's output buffer (e.g., `CLEAR 710`).

2. Clear the event registers with the `*CLS` (clear status) command.

3. Enable the "operation complete" bit (bit 0) in the Standard Event register by executing the `*ESE 1` command.

4. Send the `*OPC?` (operation complete query) command and enter the result to ensure synchronization.

5. Execute your command string to program the desired configuration, and then execute the `*OPC` (operation complete) command as the last command. When the command sequence is completed, the "operation complete" bit (bit 0) is set in the Standard Event register.

6. Use a serial poll to check to see when bit 5 (standard event) is set in the Status Byte summary register. You could also configure the function generator for an SRQ interrupt by sending `*SRE 32` (Status Byte enable register, bit 5).

The Standard Event Register

The Standard Event register reports the following types of instrument events: power-on detected, command syntax errors, command execution errors, self-test or calibration errors, query errors, or when an `*OPC` command is executed. Any or all of these conditions can be reported in the Standard Event summary bit through the *enable register*. To set the enable register mask, you write a decimal value to the register using the `*ESE` (event status enable) command.

> *An error condition (Standard Event register bits 2, 3, 4, or 5) will always record one or more errors in the function generator's error queue. Read the error queue using the* `SYSTem:ERRor?` *command.*

Bit Definitions – Standard Event Register

Bit	Decimal Value	Definition
0 Operation Complete	1	All commands prior to and including an *OPC command have been executed.
1 Not Used	2	Always set to 0.
2 Query Error	4	The function generator tried to read the output buffer but it was empty. Or, a new command line was received before a previous query has been read. Or, both the input and output buffers are full.
3 Device Error	8	A self-test or calibration error occurred (see error numbers 501 through 857 in chapter 5).
4 Execution Error	16	An execution error occurred (see error numbers -211 through -224 in chapter 5).
5 Command Error	32	A command syntax error occurred (see errors -101 through -168 in chapter 5).
6 Not Used	64	Always set to 0.
7 Power On	128	Power has been turned off and on since the last time the event register was read or cleared.

The Standard Event register is cleared when:

- You execute the `*CLS` (clear status) command.
- You query the event register using the `*ESR?` (event status register) command.

The Standard Event enable register is cleared when:

- You execute the `*ESE 0` command.
- You turn on the power and have previously configured the function generator using the `*PSC 1` command.
- The enable register *will not* be cleared at power-on if you have previously configured the function generator using `*PSC 0`.

Status Reporting Commands

SYSTem:ERRor?
Read one error from the *error queue.* When the front-panel **ERROR** annunciator turns on, one or more command syntax or hardware errors have been detected. A record of up to 20 errors can be stored in the function generator's error queue. *See chapter 5, "Error Messages," for a complete listing of the errors.*

- Errors are retrieved in first-in-first-out (FIFO) order. The first error returned is the first error that was stored. When you have read all errors from the queue, the **ERROR** annunciator turns off. The function generator beeps once each time an error is generated.
- If more than 20 errors have occurred, the last error stored in the queue (the most recent error) is replaced with *-350, "Too many errors"*. No additional errors are stored until you remove errors from the queue. If no errors have occurred when you read the error queue, the function generator responds with *+0, "No error"*.
- The error queue is cleared when power has been off or after a `*CLS` (clear status) command has been executed. The `*RST` (reset) command *does not* clear the error queue.

***CLS**
Clear the Status Byte summary register and all event registers.

***ESE** <*enable value*>
Enable bits in the Standard Event enable register. The selected bits are then reported to the Status Byte.

***ESE?**
Query the Standard Event enable register. The function generator returns a decimal value which corresponds to the binary-weighted sum of all bits set in the register.

***ESR?**
Query the Standard event register. The function generator returns a decimal value which corresponds to the binary-weighted sum of all bits set in the register.

***OPC**
Set the "operation complete" bit (bit 0) in the Standard Event register after the previous commands have been executed. *Used only in the triggered burst mode and triggered sweep mode.*

***OPC?**
Return "1" to the output buffer after the previous commands have been executed. *Used only in the triggered burst mode and triggered sweep mode.*

***PSC {0|1}**
Power-on status clear. Clear the Status Byte and Standard Event register enable masks when power is turned on (`*PSC 1`). When `*PSC 0` is in effect, the Status Byte and Standard Event register enable masks *are not* cleared when power is turned on. *The factory setting is "1" (enabled).* [*Stored in non-volatile memory*]

***PSC?**
Query the power-on status clear setting. Returns "0" (`*PSC 0`) or "1" (`*PSC 1`).

SRE <enable value***>**
Enable bits in the Status Byte enable register.

***SRE?**
Query the Status Byte enable register. The function generator returns a decimal value which corresponds to the binary-weighted sum of all bits set in the register.

***STB?**
Query the Status Byte summary register. The `*STB?` command is similar to a serial poll but it is processed like any other instrument command. The `*STB?` command returns the same result as a serial poll but the "request service" bit (bit 6) *is not* cleared if a serial poll has occurred.

***WAI**
Wait for all pending operations to complete before executing any additional commands over the interface. *Used only in the triggered burst mode and triggered sweep mode.*

An Introduction to the SCPI Language

SCPI (*Standard Commands for Programmable Instruments*) is an ASCII-based instrument command language designed for test and measurement instruments. *Refer to "Simplified Programming Overview," starting on page 136, for an introduction to the basic techniques used to program the function generator over the remote interface.*

SCPI commands are based on a hierarchical structure, also known as a *tree system*. In this system, associated commands are grouped together under a common node or root, thus forming *subsystems*. A portion of the SOURCE subsystem is shown below to illustrate the tree system.

```
SOURce:
     FREQuency:
          STARt {<frequency>|MINimum|MAXimum}
     FREQuency:
          STARt? [MINimum|MAXimum]
     FREQuency:
          STOP {<frequency>|MINimum|MAXimum}
     FREQuency:
          STOP? [MINimum|MAXimum]

     SWEep:
          SPACing {LINear|LOGarithmic}
     SWEep:
          SPACing?
     SWEep:
          TIME {<seconds>|MINimum|MAXimum}
     SWEep:
          TIME? [MINimum|MAXimum]
     SWEep:
          STATe {OFF|ON}
     SWEep:
          STATe?
```

SOURce is the root keyword of the command, FREQuency and SWEep are second-level keywords, and STARt and STOP are third-level keywords. A *colon* (**:**) separates a command keyword from a lower-level keyword.

Command Format Used in This Manual

The format used to show commands in this manual is shown below:

```
FREQuency {<frequency>|MINimum|MAXimum}
```

The command syntax shows most commands (and some parameters) as a mixture of upper- and lower-case letters. The upper-case letters indicate the abbreviated spelling for the command. For shorter program lines, send the abbreviated form. For better program readability, send the long form.

For example, in the above syntax statement, `FREQ` and `FREQUENCY` are both acceptable forms. You can use upper- or lower-case letters. Therefore, `FREQUENCY`, `freq`, and `Freq` are all acceptable. Other forms, such as `FRE` and `FREQUEN`, will generate an error.

Braces (**{ }**) enclose the parameter choices for a given command string. The braces are not sent with the command string.

A *vertical bar* (**|**) separates multiple parameter choices for a given command string.

Triangle brackets (**< >**) indicate that you must specify a value for the enclosed parameter. For example, the above syntax statement shows the *frequency* parameter enclosed in triangle brackets. The brackets are not sent with the command string. You must specify a value for the parameter (such as `"FREQ 5000"`).

Some parameters are enclosed in *square brackets* (**[ ]**). The brackets indicate that the parameter is optional and can be omitted. The brackets are not sent with the command string. If you do not specify a value for an optional parameter, the function generator chooses a default value.

Command Separators

A *colon* (**:**) is used to separate a command keyword from a lower-level keyword. You must insert a *blank space* to separate a parameter from a command keyword. If a command requires more than one parameter, you must separate adjacent parameters using a *comma* as shown below:

```
"APPL:SIN  5 KHZ, 3.0 VPP, -2.5 V"
```

A *semicolon* (**;**) is used to separate commands within the *same* subsystem, and can also minimize typing. For example, sending the following command string:

```
"FREQ:START 10; STOP 1000; MODE SWEEP"
```

... is the same as sending the following three commands:

```
"FREQ:START 10"
"FREQ:STOP 1000"
"FREQ:MODE SWEEP"
```

Use a colon *and* a semicolon to link commands from *different* subsystems. For example, in the following command string, an error is generated if you do not use both the colon *and* semicolon:

```
"SWE:STAT ON;:TRIG:SOUR EXT"
```

Using the *MIN* and *MAX* Parameters

You can substitute `MINimum` or `MAXimum` in place of a parameter for many commands. For example, consider the following command:

```
FREQuency {<frequency>|MINimum|MAXimum}
```

Instead of selecting a specific frequency, you can substitute `MIN` to set the frequency to its minimum value or `MAX` to set the frequency to its maximum value.

Querying Parameter Settings

You can query the current value of most parameters by adding a *question mark* (**?**) to the command. For example, the following command sets the output frequency to 5 kHz:

```
"FREQ 5000"
```

You can query the frequency value by executing:

```
"FREQ?"
```

You can also query the minimum or maximum frequency allowed with the present function as follows:

```
"FREQ? MIN"
"FREQ? MAX"
```

SCPI Command Terminators

A command string sent to the function generator *must* terminate with a <*new line*> character. The IEEE-488 *EOI* (end-or-identify) message is interpreted as a <*new line*> character and can be used to terminate a command string in place of a <*new line*> character. A <*carriage return*> followed by a <*new line*> is also accepted. Command string termination will *always* reset the current SCPI command path to the root level.

IEEE-488.2 Common Commands

The IEEE-488.2 standard defines a set of *common commands* that perform functions like reset, self-test, and status operations. Common commands always begin with an asterisk (*), are four to five characters in length, and may include one or more parameters. The command keyword is separated from the first parameter by a *blank space*. Use a *semicolon* (**;**) to separate multiple commands as shown below:

```
"*RST; *CLS; *ESE 32; *OPC?"
```

SCPI Parameter Types

The SCPI language defines several different data formats to be used in program messages and response messages.

Numeric Parameters Commands that require numeric parameters will accept all commonly used decimal representations of numbers including optional signs, decimal points, and scientific notation. Special values for numeric parameters like `MINimum`, `MAXimum`, and `DEFault` are also accepted. You can also send engineering unit suffixes with numeric parameters (e.g., `Mhz` or `Khz`). If only specific numeric values are accepted, the function generator will automatically round the input numeric parameters. The following command uses a numeric parameter:

```
FREQuency {<frequency>|MINimum|MAXimum}
```

Discrete Parameters Discrete parameters are used to program settings that have a limited number of values (like `BUS`, `IMMediate`, `EXTernal`). They have a short form and a long form just like command keywords. You can mix upper- and lower-case letters. Query responses will *always* return the short form in all upper-case letters. The following command uses discrete parameters:

```
SWEep:SPACing {LINear|LOGarithmic}
```

Boolean Parameters Boolean parameters represent a single binary condition that is either true or false. For a false condition, the function generator will accept “OFF” or “0”. For a true condition, the function generator will accept “ON” or “1”. When you query a boolean setting, the instrument will *always* return “0” or “1”. The following command uses a boolean parameter:

```
AM:STATe {OFF|ON}
```

String Parameters String parameters can contain virtually any set of ASCII characters. A string *must* begin and end with matching quotes; either with a single quote or with a double quote. You can include the quote delimiter as part of the string by typing it twice without any characters in between. The following command uses a string parameter:

```
DISPlay:TEXT <quoted string>
```

Halting an Output in Progress

You can send a *device clear* at any time to stop an output in progress over the GPIB interface. The status registers, the error queue, and all configuration states are left unchanged when a device clear message is received. Device clear performs the following actions:

- The function generator's input and output buffers are cleared.
- The function generator is prepared to accept a new command string.
- The following statement shows how to send a device clear over the GPIB interface:

```
CLEAR 710        IEEE-488 Device Clear
```

- The following statement shows how to send a device clear over the GPIB interface using the GPIB Command Library for *C* or *QuickBASIC*:

```
IOCLEAR (710)
```

For RS-232 operation, sending the <Ctrl-C> character will perform the same operation as the IEEE-488 device clear message. The function generator's DTR (data terminal ready) handshake line is set "true" following a device clear message. See "DTR/DSR Handshake Protocol," on page 198 for further details.

To set the GPIB address

Each device on the GPIB (IEEE-488) interface must have a unique address. You can set the function generator's address to any value between 0 and 30. *The address is set to **10** when the function generator is shipped from the factory*. The current address is displayed momentarily on the front panel when you turn on the function generator. *See also "GPIB Address," on page 114.*

Shift
Menu On/Off

1 Turn on the menu.

```
A: MOD MENU
```

< <

2 Move across to the I/O MENU choice on this level.

```
E: I/O MENU
```

∨

3 Move down a level to the HPIB ADDR command.

```
1: HPIB ADDR
```

∨

4 Move down to the "PARAMETER" level to set the address.

Use the left/right and down/up arrow keys to change the address.

```
∧10      ADDR
```

Enter

5 Save the change and turn off the menu.

The address is stored in *non-volatile* memory, and *does not* change when power has been off or after a remote interface reset.

To select the remote interface

The function generator is shipped with both an GPIB (IEEE-488) interface and an RS-232 interface. Only one interface can be enabled at a time. *The **GPIB interface** is selected when the function generator is shipped from the factory*. *See also "Remote Interface Selection," on page 115.*

Shift
Menu On/Off

1 Turn on the menu.

```
A: MOD MENU
```

< <

2 Move across to the I/O MENU choice on this level.

```
E: I/O MENU
```

v >

3 Move down a level and then across to the INTERFACE choice.

```
2: INTERFACE
```

v

4 Move down to the "PARAMETER" level to select the interface.

Use the left/right arrow keys to see the interface choices. Choose from the following: HPIB / 488 or RS-232.

```
HPIB / 488
```

Enter

5 Save the change and turn off the menu.

The interface selection is stored in *non-volatile* memory, and *does not* change when power has been off or after a remote interface reset.

To set the baud rate

You can select one of six baud rates for RS-232 operation. *The rate is set to **9600 baud** when the function generator is shipped from the factory. See also "Baud Rate Selection," on page 116.*

Shift
Menu On/Off

1 Turn on the menu.

```
A: MOD MENU
```

< <

2 Move across to the I/O MENU choice on this level.

```
E: I/O MENU
```

∨ > >

3 Move down a level and then across to the BAUD RATE choice.

```
3: BAUD RATE
```

∨

4 Move down to the "PARAMETER" level to select the baud rate.

Use the left/right arrow keys to see the baud rate choices. Choose from one of the following: 300, 600, 1200, 2400, 4800, or **9600** baud.

```
9600  BAUD
```

Enter

5 Save the change and exit the menu.

The baud rate selection is stored in *non-volatile* memory, and *does not* change when power has been off or after a remote interface reset.

To set the parity

You can select the parity for RS-232 operation. *The function generator is configured for **8 data bits with no parity** when shipped from the factory. See also "Parity Selection," on page 116.*

Shift
Menu On/Off

1 Turn on the menu.

```
A: MOD MENU
```

< <

2 Move across to the I/O MENU choice on this level.

```
E: I/O MENU
```

∨ < <

3 Move down a level and then across to the PARITY choice.

```
4: PARITY
```

∨

4 Move down to the "PARAMETER" level to select the parity.

Use the left/right arrow keys to see the parity choices. Choose from one of the following: **None: 8 Bits**, Odd: 7 Bits, or Even: 7 bits. When you set parity, you are indirectly setting the number of data bits.

```
NONE: 8 BITS
```

Enter

5 Save the change and turn off the menu.

The parity selection is stored in *non-volatile* memory, and *does not* change when power has been off or after a remote interface reset.

SCPI Conformance Information

The Agilent 33120A Function Generator/Arbitrary Waveform Generator conforms to the 1993.0 version of the SCPI standard. Many of the commands required by the standard are accepted by the function generator but are not described in this manual for simplicity or clarity. Most of these non-documented commands duplicate the functionality of a command already described in this manual.

SCPI Confirmed Commands The following table lists the SCPI-approved commands that are used by the function generator.

SCPI Confirmed Commands

```
CALibration
  [:ALL]?
  :VALue <value>
  :VALue?

DATA
  :CATalog?
  :COPY <destination arb name> [,VOLATILE]
  [:DATA] VOLATILE, <value>,<value>, . . .
  :DELete:ALL
  :DELete[:NAME] <arb name>

DISPlay
  [WINDow][:STATe] {OFF|ON}
  [WINDow][:STATe]?
  [WINDow]:TEXT:CLEar
  [WINDow]:TEXT[:DATA] <quoted string>
  [WINDow]:TEXT[:DATA]?

FORMat
  :BORDer {NORMal|SWAPped}
  :BORDer?

[SOURce]
  :AM:DEPTh {<depth in percent>|MINimum|MAXimum}
  :AM:DEPTh? [MINimum|MAXimum]
  :AM:INTernal:FREQuency {<frequency>|MINimum|MAXimum}
  :AM:INTernal:FREQuency? [MINimum|MAXimum]
  :AM:SOURce EXTernal
  :AM:SOURce?
```

SCPI Confirmed Commands *(continued)*

```
[SOURce]
  :AM:STATe {OFF|ON}
  :AM:STATe?
  :FM:DEViation {<peak deviation in Hz>|MINimum|MAXimum}
  :FM:DEViation? [MINimum|MAXimum]
  :FM:INTernal:FREQuency {<frequency>|MINimum|MAXimum}
  :FM:INTernal:FREQuency? [MINimum|MAXimum]
  :FM:STATe {OFF|ON}
  :FM:STATe?
  :FREQuency {<frequency>|MINimum|MAXimum}
  :FREQuency? [MINimum|MAXimum]
  :FREQuency:CENTer
  :FREQuency:MODE
  :FREQuency:SPAN
  :FREQuency:STARt {<frequency>|MINimum|MAXimum}
  :FREQuency:STARt? [MINimum|MAXimum]
  :FREQuency:STOP {<frequency>|MINimum|MAXimum}
  :FREQuency:STOP? [MINimum|MAXimum]
  :FUNCtion:SHAPe {SINusoid|SQUare|TRIangle|RAMP|NOISe|DC|USER}
  :FUNCtion:SHAPe?
  :PULSe:DCYCle {<percent>|MINimum|MAXimum}
  :PULSe:DCYCle? [MINimum|MAXimum]
  :SWEep:SPACing {LINear|LOGarithmic}
  :SWEep:SPACing?
  :SWEep:TIME {<seconds>|MINimum|MAXimum}
  :SWEep:TIME? [MINimum|MAXimum]
  :VOLTage[:LEVel][:IMMediate]...
      [:AMPLitude] {<amplitude>|MINimum|MAXimum}
  :VOLTage[:LEVel][:IMMediate][:AMPLitude]? [MINimum|MAXimum]
  :VOLTage[:LEVel][:IMMediate]:OFFSet {<offset>|MINimum|MAXimum}
  :VOLTage[:LEVel][:IMMediate]:OFFSet? [MINimum|MAXimum]
  :VOLTage:UNIT {VPP|VRMS|DBM|DEFault}
  :VOLTage:UNIT?

SYSTem
  :BEEPer[IMMediate]
  :ERRor?
  :VERSion?

TRIGger
  [SEQuence]:SOURce {IMMediate|EXTernal|BUS}
  [SEQuence]:SOURce?
```

Device-Specific Commands The following commands are designed specifically for the 33120A. They are not included in the 1993.0 version of the SCPI standard. However, these commands are designed with the SCPI standard in mind and they follow all of the command syntax rules defined by the standard.

Non-SCPI Commands

```
CALibration
  :COUNt?
  :SECure:CODE <new code>
  :SECure:STATe {OFF|ON},<code>
  :SECure:STATe?
  :SETup <0|1|2|3| ... |84>
  :SETup?
  :STRing <quoted string>
  :STRing?

DATA
  :ATTRibute:AVERage? <arb name>
  :ATTRibute:CFACtor? <arb name>
  :ATTRibute:POINts? <arb name>
  :ATTRibute:PTPeak? <arb name>
  :DAC VOLATILE, {<binary block>|<value>,<value>, . . .}
  :NVOLatile:CATalog?
  :NVOLatile:FREE?

OUTPut
  :LOAD {50|INFinity|MINimum|MAXimum}
  :LOAD? [MINimum|MAXimum]
  :SYNC {OFF|ON}
  :SYNC?

MEMory:STATe:DELete {0|1|2|3}

[SOURce]
  :AM:INTernal:FUNCtion {SINusoid|SQUare|TRIangle|RAMP|NOISe|USER}
  :AM:INTernal:FUNCtion?
  :APPLy:SINusoid [<frequency> [,<amplitude> [,<offset>] ]]
  :APPLy:SQUare [<frequency> [,<amplitude> [,<offset>] ]]
  :APPLy:TRIangle [<frequency> [,<amplitude> [,<offset>] ]]
  :APPLy:RAMP [<frequency> [,<amplitude> [,<offset>] ]]
  :APPLy:NOISe [<frequency|DEFault> [,<amplitude> [,<offset>] ]]
  :APPLy:DC [<frequency|DEFault> [,<amplitude|DEFault> [,<offset>] ]]
  :APPLy:USER [<frequency> [,<amplitude> [,<offset>] ]]
  :APPLy?
```

Non-SCPI Commands *(continued)*

```
[SOURce]
  :BM:INTernal:RATE {<frequency>|MINimum|MAXimum}
  :BM:INTernal:RATE? [MINimum|MAXimum]
  :BM:NCYCles {<# cycles>|INFinity|MINimum|MAXimum}
  :BM:NCYCles? [MINimum|MAXimum]
  :BM:PHASe {<degrees>|MINimum|MAXimum}
  :BM:PHASe? [MINimum|MAXimum]
  :BM:SOURce {INTernal|EXTernal}
  :BM:SOURce?
  :BM:STATe {OFF|ON}
  :BM:STATe?
  :FM:INTernal:FUNCtion {SINusoid|SQUare|TRIangle|RAMP|NOISe|USER}
  :FM:INTernal:FUNCtion?
  :FSKey:FREQuency {<frequency>|MINimum|MAXimum}
  :FSKey:FREQuency? [MINimum|MAXimum]
  :FSKey:INTernal:RATE {<rate in Hz>|MINimum|MAXimum}
  :FSKey:INTernal:RATE? [MINimum|MAXimum]
  :FSKey:SOURce {INTernal|EXTernal}
  :FSKey:SOURce?
  :FSKey:STATe {OFF|ON}
  :FSKey:STATe?
  :FUNCtion:USER {<arb name>|VOLATILE}
  :FUNCtion:USER?
  :SWEep:STATe {OFF|ON}
  :SWEep:STATe?

SYSTem:
  :LOCal
  :REMote
  :RWLock
```

IEEE-488 Conformance Information

Dedicated Hardware Lines

ATN	*Attention*
IFC	*Interface Clear*
REN	*Remote Enable*
SRQ	*Service Request Enable*

Addressed Commands

DCL	*Device Clear*
EOI	*End or Identify Terminator*
GET	*Group Execute Trigger*
GTL	*Go To Local*
LLO	*Local Lockout*
SDC	*Selected Device Clear*
SPD	*Serial Poll Disable*
SPE	*Serial Poll Enable*

Capability Identification Codes

```
SH1, AH1, T6, L4, SR1, RL1,
PP0, DC1, DT1, C0, E2
```

IEEE-488.2 Common Commands

```
*CLS
*ESE <enable value>
*ESE?
*ESR?
*IDN?
*OPC
*OPC?
*PSC {0|1}
*PSC?
*RST
*SAV {0|1|2|3}
*RCL {0|1|2|3}
*SRE <enable value>
*SRE?
*STB?
*TRG
*TST?
*WAI
```

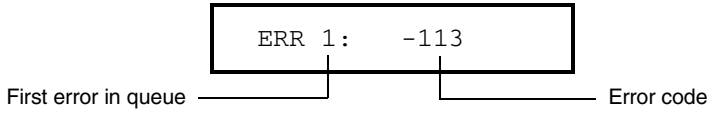
5

Error Messages

- Errors are retrieved in first-in-first-out (FIFO) order. The first error returned is the first error that was stored. When you have read all errors from the queue, the **ERROR** annunciator turns off. The function generator beeps once each time an error is generated.
- If more than 20 errors have occurred, the last error stored in the queue (the most recent error) is replaced with *-350, "Too many errors"*. No additional errors are stored until you remove errors from the queue. If no errors have occurred when you read the error queue, the function generator responds with *+0, "No error"*.
- The error queue is cleared when power has been off or after a `*CLS` (clear status) command has been executed. The `*RST` (reset) command *does not* clear the error queue.
- *Front-Panel Operation*:

  `3: ERROR` **`(SYS MENU)`**

  If the **ERROR** annunciator is on, press Shift < (Recall Menu) to read the errors stored in the queue. The errors are listed *horizontally* on the "PARAMETER" level. All errors are cleared when you go to the "PARAMETER" level and then turn off the menu.



- *Remote Interface Operation*:

  `SYSTem:ERRor?` *Reads one error from the error queue*

  Errors have the following format (the error string may contain up to *80 characters*):

  ```
  -113,"Undefined header"
  ```

Execution Errors

-101 **Invalid character**
An invalid character was found in the command string. You may have inserted a character such as #, $, or % in the command header or within a parameter. Example: `TRIG:SOUR BUS#`

-102 **Syntax error**
Invalid syntax was found in the command string. You may have inserted a blank space before or after a colon (or before a comma) in the command header. Example: `APPL:SIN  ,1`

-103 **Invalid separator**
An invalid separator was found in the command string. You may have used a comma instead of a colon, semicolon, or blank space – or you may have used a blank space instead of a comma. Example: `TRIG:SOUR,BUS` or `APPL:SIN 1  1000`

-105 **GET not allowed**
A Group Execute Trigger (`GET`) is not allowed within a command string.

-108 **Parameter not allowed**
More parameters were received than were expected for the command. You may have entered an extra parameter, or you added a parameter to a command that does not accept a parameter. Example: `APPL? 10`

-109 **Missing parameter**
Fewer parameters were received than were expected for the command. You omitted one or more parameters that are required for the command. Example: `OUTP:LOAD`

-112 **Program mnemonic too long**
A command header was received which contained more than the maximum 12 characters allowed. Example: `OUTP:SYNCHRONIZATION ON`

-113 **Undefined header**
A command was received that is not valid for the function generator. You may have misspelled the command or it may not be a valid command. If you are using the short form of the command, remember that it may contain up to four letters. Example: `TRIGG:SOUR BUS`

-121 **Invalid character in number**
An invalid character was found in the number specified for a parameter value. Example: `*ESE #B01010102`

-123 **Exponent too large**
A numeric parameter was found whose exponent was larger than 32,000. Example: `BMOD:NCYC 1E34000`

-124 **Too many digits**
A numeric parameter was found whose mantissa contained more than 255 digits, excluding leading zeros.

-128 **Numeric data not allowed**
A numeric parameter was received but a character string was expected. Example: `DISP:TEXT 123`

-131 **Invalid suffix**
A suffix was incorrectly specified for a numeric parameter. You may have misspelled the suffix. Example: `SWE:TIME 0.5 SECS`

-138 **Suffix not allowed**
A suffix was received following a numeric parameter which does not accept a suffix. Example: `BM:PHAS 10 DEG` (this command does not allow a suffix).

-148 **Character data not allowed**
A discrete parameter was received but a character string or a numeric parameter was expected. Check the list of parameters to verify that you have used a valid parameter type. Example: `DISP:TEXT ON`

-151 **Invalid string data**
An invalid character string was received. Check to see if you have enclosed the character string in single or double quotes and that the string contains valid ASCII characters.

-158 **String data not allowed**
A character string was received but is not allowed for the command. Check the list of parameters to verify that you have used a valid parameter type. Example: `BMOD:NCYC 'TEN'`

-161 **Invalid block data**
Applies only to the `DATA:DAC VOLATILE` command. For a definite-length block, the number of bytes of data sent does not match the number of bytes that you specified in the block header. For an indefinite-length block, an *EOI* (End-or-Identify) was received without an accompanying <*new line*> character.

-168 **Block data not allowed**
Block data was received but is not allowed for the command. Check to see if you have sent the correct data type with the command.

-170 to -178 **Expression errors**
The function generator does not accept mathematical expressions.

-211 **Trigger ignored**
A Group Execute Trigger (`GET`) or `*TRG` was received but the trigger was ignored. Make sure that you have selected the correct trigger source.

-221 **Settings conflict; *amplitude* has been adjusted**
The requested output amplitude is not valid. The output amplitude is automatically adjusted to be compatible with the present configuration. This error will be generated in the following instances:

- The output amplitude is automatically adjusted if you select a function whose maximum amplitude is *less than* that of the currently active function. This conflict may arise when the output units are *Vrms* or *dBm* due to the differences in crest factor.
- If the specified amplitude is not valid, the function generator will automatically adjust it to the maximum value allowed with the present offset voltage. The output amplitude (in V*pp*) and the dc offset voltage must obey the following restrictions. (V*max* is either 10 volts for a high impedance termination or 5 volts for a 50 ohm termination.)

$$|\,V_{offset}\,| + \frac{V_{pp}}{2} \le V_{max} \quad and \quad |\,V_{offset}\,| \le 2 \times V_{pp}$$

-221 **Settings conflict; cannot adjust phase in present configuration**
Option 001 Phase-Lock Only. The phase cannot be adjusted real-time if an arbitrary waveform is selected, a modulation mode (other that burst) is enabled, or if burst is enabled with a burst count other than infinity.

-221 **Settings conflict; *duty cycle* has been adjusted**
The requested duty cycle is not valid. For output frequencies above 5 MHz, the duty cycle is limited to values between 40% and 60% (below 5 MHz, the range is 20% to 80%). The duty cycle is automatically adjusted if you select a frequency that is not valid with the present duty cycle percentage. For example, if you set the duty cycle to 70% and then change the frequency to 8 MHz, the function generator will automatically adjust the duty cycle to 60% (the upper limit for this frequency).

-221 **Settings conflict; *fm deviation* has been adjusted**
The requested FM *peak frequency deviation* is not valid. The sum of the *carrier frequency* and peak frequency deviation must be less than or equal to the maximum frequency for the selected function ***plus 100 kHz*** (15.1 MHz for sine and square, 200 kHz for triangle and ramp, and 5.1 MHz for arbitrary waveforms). If you attempt to set the deviation to a value that is not valid, the function generator will automatically adjust the deviation to the maximum value allowed with the present carrier frequency.

-221 **Settings conflict; *frequency* has been adjusted**
The requested output (or carrier) frequency is not valid. The output frequency is automatically adjusted if you select a function whose maximum frequency is *less than* that of the currently active function. For example, if you output a 1 MHz sine wave and then change the function to triangle wave, the function generator will adjust the output to 100 kHz (the upper limit for triangle waves).

-221 **Settings conflict; *fsk frequency* has been adjusted**
The requested FSK "hop" frequency is not valid. The FSK frequency is automatically adjusted if you select a function whose maximum frequency is *less than* that of the currently active function. For example, if you set the FSK frequency to 1 MHz in the sine function and then change the function to triangle wave, the function generator will adjust the frequency to 100 kHz (the upper limit for triangle waves).

-221 **Settings conflict; *offset* has been adjusted**
The requested offset voltage is not valid with the present output amplitude. The offset is automatically adjusted to the maximum value allowed with the present output amplitude. The output amplitude (in V*pp*) and the dc offset voltage must obey the following restrictions. (V*max* is either 10 volts for a high impedance termination or 5 volts for a 50 ohm termination.)

$$|V_{offset}| + \frac{V_{pp}}{2} \leq V_{max} \quad and \quad |V_{offset}| \leq 2 \times V_{pp}$$

-221 **Settings conflict; *start frequency* has been adjusted**
The requested start frequency is not valid. The start frequency is automatically adjusted if you select a function whose maximum frequency is *less than* that of the currently active function. For example, if you set the start frequency to 200 kHz in the sine wave function and then change the function to triangle wave, the function generator will adjust the output to 100 kHz (the upper limit for triangle waves).

-221 **Settings conflict; *stop frequency* has been adjusted**
The requested stop frequency is not valid. The stop frequency is automatically adjusted if you select a function whose maximum frequency is *less than* that of the currently active function. For example, if you set the stop frequency to 1 MHz in the sine wave function and then change the function to triangle wave, the function generator will adjust the output to 100 kHz (the upper limit for triangle waves).

-221 **Settings conflict; previous *modulation* has been disabled**
Only one modulation state can be enabled at a time. When you enable a modulation state, the previous modulation state is turned off. For example, if AM is on and then you enable FM, the function generator will turn off AM first.

-222 **Data out of range**
A numeric parameter value is outside the valid range for the command. Example: `BMOD:NCYC -3` or `FREQ 16 MHZ`

-222 **Data out of range; *amplitude***
The requested output amplitude exceeds the upper limit for the selected function. This error applies only to the `APPLy` command. *See "Output Amplitude" on page 59 for more information.*

-222 **Data out of range; *frequency***
The requested output (or carrier) frequency exceeds the upper limit for the selected function. This error applies only to the `APPLy` command. *See "Output Frequency" on page 57 for more information.*

-222 **Data out of range; *offset***
The requested offset voltage exceeds the upper limit for the selected function or output amplitude. This error applies only to the `APPLy` command. *See "DC Offset Voltage" on page 62 for more information.*

-223 **Too much data**
A character string was received but could not be executed because the string length was more than 60 characters. This error can be generated by the `CALibration:STRing` and `DISPlay:TEXT` commands.

-224 **Illegal parameter value**
A discrete parameter was received which was not a valid choice for the command. You may have used an invalid parameter choice. Example: `DISP:STAT XYZ` (**XYZ** is not a valid choice).

-330 **Self-test failed**
The function generator's self-test failed from the remote interface (`*TST?` command). In addition to this error, one or more specific errors may be reported. *See also "Self-Test Errors," on page 237.*

-350 **Too many errors**
The error queue is full because more than 20 errors have occurred. No additional errors are stored until you remove errors from the queue. The error queue is cleared when power has been off, or after a `*CLS` (clear status) command has been executed.

-410 **Query INTERRUPTED**
A command was received which sends data to the output buffer, but the output buffer contained data from a previous command (the previous data is not overwritten). The output buffer is cleared when power has been off, or after a device clear has been executed.

-420 **Query UNTERMINATED**
The function generator was addressed to talk (i.e., to send data over the interface) but a command has not been received which sends data to the output buffer. For example, you may have executed an `APPLy` command (which does not generate data) and then attempted an `ENTER` statement to read data from the remote interface.

-430 **Query DEADLOCKED**
A command was received which generates too much data to fit in the output buffer and the input buffer is also full. Command execution continues but all data is lost.

-440 **Query UNTERMINATED after indefinite response**
The `*IDN?` command must be the last query command within a command string. Example: `*IDN?;:SYST:VERS?`

501 **Isolator UART framing error**

502 **Isolator UART overrun error**

511 **RS-232 framing error**

512 **RS-232 overrun error**

513 **RS-232 parity error**

514 **Command allowed only with RS-232**
The following three commands are used only with RS-232: `SYSTem:LOCal`, `SYSTem:REMote`, and `SYSTem:RWLock`.

521 **Input buffer overflow**

522 **Output buffer overflow**

550 **Command not allowed in local**
You should always execute the `SYSTem:REMote` command before sending other commands over the RS-232 interface. This error is also generated if you attempt to execute the `DISP:TEXT` command while in the local mode (this command is allowed in remote only).

580 **Phase-locked loop is unlocked**
Option 001 Phase-Lock Only. The function generator has detected an "unlock" condition. You must execute the `PHAS:UNL:ERR:STAT ON` command to enable this error.

800 **Block length must be even**
The function generator represents binary data for arbitrary waveforms as a 12-bit integers, which is sent as *two* bytes (`DATA:DAC VOLATILE` command only). An odd number of bytes is not accepted because the function generator would not know how to interpret the last single byte.

810 **State has not been stored**
The memory location specified in the `*RCL` command was not used in a previous `*SAV` command. You cannot recall the instrument state from a memory location that was not previously specified as a storage location.

Self-Test Errors

The following errors indicate failures that may occur during a self-test. Refer to the *Service Guide* for more information.

590 **I/O processor reset; possible low power line voltage**

601 **Front panel does not respond**

602 **RAM read/write failed**

603 **Waveform RAM readback failed**

604 **Modulation RAM readback failed**

605 **Serial configuration readback failed**

606 **Waveform ASIC failed**

607 **SYNC signal detection failure, Bessel filter path**

608 **SYNC signal detection failure, Elliptic filter path**

625 **I/O processor does not respond**

626 **I/O processor failed self-test**

Calibration Errors

The following errors indicate failures that may occur during a calibration. Refer to the *Service Guide* for more information.

701 **Cal security disabled by jumper**
The calibration security feature has been disabled with a jumper inside the function generator. When applicable, this error will occur at power-on to warn you that the function generator is unsecured.

702 **Cal secured**
The function generator is secured against calibration.

703 **Invalid secure code**
An invalid calibration security code was received when attempting to unsecure or secure the function generator. You must use the same security code to unsecure the function generator as was used to secure it, and vice versa. The security code may contain up to 12 alphanumeric characters. The first character *must be* a letter.

704 **Secure code too long**
A security code was received which contained more than 12 characters.

705 **Cal aborted**
A calibration in progress is aborted when you press any front-panel key, send a device clear, or change the local/remote state of the instrument.

706 **Cal value out of range**
The specified calibration value (`CAL:VALUE`) is invalid for the presently selected function.

707 **Cal signal measurement out of range**
The specified calibration value (`CAL:VALUE`) does not match the signal applied to the function generator.

708 **Flatness cal failed**

709 **Cannot calibrate frequency while externally locked**
You cannot perform `CAL:SETUP 0` or `CAL:SETUP 86` while the function generator is externally phase locked (*Option 001*).

760 **RAM checksum failure**

770 **Nonvolatile arb waveform memory checksum failure**

771 **Nonvolatile cal memory checksum failure**

772 **Nonvolatile system memory checksum failure**

773 **Nonvolatile state memory checksum failure**

774 **Nonvolatile memory erase failure**

775 **Nonvolatile memory write failure**

780 to 788 *See "Arbitrary Waveform Errors" on page 240.*

800 **Block length must be even**
The function generator represents binary data for arbitrary waveforms as a 12-bit integers, which is sent as *two* bytes (`DATA:DAC VOLATILE` command only). An odd number of bytes is not accepted because the function generator would not know how to interpret the last single byte.

810 **State has not been stored**
The memory location specified in the `*RCL` command was not used in a previous `*SAV` command. You cannot recall the instrument state from a memory location that was not previous specified as a storage location.

850 **Cal setup invalid**

851 **Negative offset gain cal required (CAL:SETup 50)**

852 **Flatness DAC gain cal required (CAL:SETup 64)**

853 **AM cal 1 required (CAL:SETup 30)**

854 **AM cal 2 required (CAL:SETup 31)**

855 **Cal load resistance not specified (CAL:SETup 33)**

856 **Square wave positive offset cal required (CAL:SETup 60)**

857 **Square wave 50% duty cycle cal required (CAL:SETup 62)**

858 **AM cal 3 required (CAL:SETup 32)**

Arbitrary Waveform Errors

The following errors indicate failures that may occur during an arbitrary waveform download. Refer to "Arbitrary Waveform Commands," starting on page 174, for more information.

780 **VOLATILE arb waveform has not been loaded**
The DATA:COPY command copies the waveform downloaded into VOLATILE memory to the specified name. You must download the waveform using the DATA VOLATILE or DATA:DAC VOLATILE command prior to a copy operation.

781 **Not enough memory to store new arb waveform; use DATA:DELete**
Up to four user-defined waveforms can be stored in non-volatile memory. Use the DATA:DEL command to delete the waveform in VOLATILE memory or any of the four user-defined waveforms in non-volatile memory. Use the DATA:CAT? command to list all waveforms currently stored in volatile and non-volatile memory.

782 **Cannot overwrite a built-in arb waveform**
You cannot overwrite the names of the five built-in arbitrary waveforms. The following names are reserved and cannot be used with the DATA:COPY command: SINC, NEG_RAMP, EXP_RISE, EXP_FALL, and CARDIAC.

783 **Arb waveform name too long**
The arb name can contain up to 8 characters (used with DATA:COPY command). The first character *must* be a letter (A-Z), but the remaining characters can be numbers (0-9) or the underscore character (" _ "). Blank spaces are not allowed.

784 **Name of source arb waveform for copy must be VOLATILE**
The DATA:COPY command copies the arbitrary waveform downloaded into VOLATILE memory to the specified name. The source for the copy operation is always "VOLATILE" (you cannot copy from any other name).

785 **Specified arb waveform does not exist**
The arb name specified with the FUNC:USER or DATA:DEL command has not been downloaded into memory. Use the DATA:CAT? command to list all waveforms currently downloaded. To download a waveform into memory, use the DATA VOLATILE or DATA:DAC VOLATILE commands.

This error may also be generated if you insert a space instead of a colon before the ALL parameter in the DATA:DEL:ALL command. *The colon before the ALL parameter is required* – otherwise, the function generator will attempt to delete the arbitrary waveform with the name "ALL".

786 **Cannot delete a built-in arb waveform**
You cannot delete any of the five built-in arbitrary waveforms. The following names are reserved and cannot be deleted with the DATA:DEL command: SINC, NEG_RAMP, EXP_RISE, EXP_FALL, and CARDIAC.

787 **Cannot delete the currently selected active arb waveform**
You cannot delete the arbitrary waveform that is currently being output. Either select a different arbitrary waveform or change to a different output function.

788 **Cannot copy to VOLATILE arb waveform**
The DATA:COPY command copies the arbitrary waveform from volatile memory to the specified name in non-volatile memory. The source for the copy operation is always "VOLATILE" (you cannot copy from any other name) and you cannot copy *to* VOLATILE.

790 **Arb waveform must contain at least 8 points**
You cannot create an arbitrary waveform with fewer than 8 data points. Redefine the waveform with between 8 and 16,000 data points.

800 **Block length must be even**
The function generator represents binary data as a 12-bit integers, which is sent as *two* bytes (DATA:DAC VOLATILE command only). An odd number of bytes is not accepted because the function generator would not know how to interpret the last single byte.

810 **State has not been stored**
The memory location specified in the *RCL command was not used in a previous *SAV command. You cannot recall the instrument state from a memory location that was not previously specified as a storage location.

6

Application Programs

Application Programs

This chapter contains several remote interface application programs to help you develop programs for your own application. Chapter 4, "Remote Interface Reference," starting on page 125, lists the syntax for the SCPI (*Standard Commands for Programmable Instruments*) commands available to program the function generator.

BASIC Language Programs

All of the BASIC example programs in this chapter were developed and tested on an HP 9000 Series 300 controller. Each device on the GPIB (IEEE-488) interface must have a unique address. You can set the function generator's address to any value between 0 and 30. The current address is displayed momentarily on the front panel when you turn on the function generator.

The GPIB (IEEE-488) address is set to "**10**" when the function generator is shipped from the factory. The example programs in this chapter assume an GPIB address of 10. When sending a command over the remote interface, you append this address to the GPIB interface's select code (normally "7"). For example, if the select code is "7" and the device address is "10", the combination is "710".

C Language Programs

All of the C Language example programs in this chapter are written for the *82335 GPIB Interface Card* using the *GPIB Command Library for C*. Unless otherwise noted, the library functions used in the example programs are compatible with the ANSI C standard.

All of the C Language programs were compiled (and tested) using the following compilers:

- Microsoft® QuickC® Version 2.0
- Borland® Turbo C®++ Version 1.0

Compiling and Linking a C Program

To run a program, first compile and link the program to make an executable file. To compile and link a program:

- Make sure the necessary PATH statements have been added to your AUTOEXEC.BAT file to enable the compiler to find the library and header files (see your C Language manual to specify the proper paths).
- Link the appropriate GPIB C library file for your compiler (the file is located on the GPIB Command Library disk that came with the GPIB Interface Card). Use the appropriate library file:

  `clhpib.lib` *for Microsoft® QuickC®*

  `tchhpib.lib` *for Borland® Turbo C®++*

- If you *are not* compiling in the "large" or "huge" memory model, include the "CFUNC.H" header file in your program (the file is located on the GPIB Command Library disk that came with the GPIB Interface Card).
- To compile Microsoft® QuickC® programs from the DOS command line (using the "large" memory model), execute the following:

  `qcl /AL <`*path...* `\`*program.C*`> <`*path...* `\clhpib.lib>`

  For example:

  ```
  qcl /AL b:\arb_wave.c  c:\qc2\lib\clhpib.lib
  ```

  Change the "`/AL`" parameter to the appropriate setting when compiling in the smaller memory models (see your C Language manual for more information). For some programs, you may also have to change the stack size using the "`/F `" parameter.

- To compile Borland® Turbo C®++ programs from the DOS command line (using the "large" memory model), execute the following:

  ```
  tcc -ml <path... \program.C> <path... \tchhpib.lib>
  ```

  For example:

  ```
  tcc -ml b:\arb_wave.c  c:\tc\lib\tchhpib.lib
  ```

  Change the "`-ml`" parameter to the appropriate setting when compiling in the smaller memory models (see your C Language manual for more information).

- Once compiled and linked, an executable file (.EXE) and object file (.OBJ) are created in the current directory. Run the program by typing the file name with the .EXE extension.

QuickBASIC Language Programs

All of the QuickBASIC Language example programs in this chapter are written for the *82335 GPIB Interface Card* using the *GPIB Command Library for BASIC*.

Running a QuickBASIC Program

To run a program, first compile and link the program to make an executable file. You can either compile and link the program in the QuickBASIC environment, or compile and link separately from the DOS command line.

- To compile and link a program in the QuickBASIC environment, execute the following:

  ```
  qb <path... \program.BAS> /l <path... \qbhpib>
  ```

  This loads both the program and the GPIB library into the QuickBASIC environment. Note that this only selects the environment for QuickBASIC versions 4.0 and above only.

- To compile and link a program from the DOS command line is a two-step method as follows:

  First, compile the program by executing the following:

  ```
  bc <path... \program.BAS>
  ```

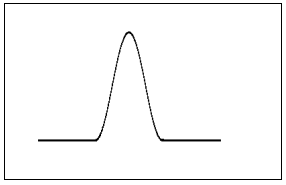
  This generates an object listing that has the same name as the program name with an ".OBJ" extension. The object file must be linked to the GPIB library.

  Second, link the GPIB library:

  ```
  link <path... \program.BAS>
  ```

  After prompted for a run file, press the Enter key. This generates the executable file with an ".EXE" extension.

Using the APPLy Command



This program teaches the following concepts:

- How to use the `APPLy` command with low-level commands to set up a burst. By setting the starting phase to 270 degrees and adding an offset voltage, a bursted "haversine" waveform is generated.
- How to set up a trigger source and trigger the function generator over the GPIB interface.

BASIC / GPIB (Program 1)

```
10 !
20 ! This program sets up a burst waveform with a 270 degree
30 ! starting phase. By adding an offset voltage to the burst,
40 ! a "haversine" is created. This program also shows the use
50 ! of a trigger received over the GPIB interface to
60 ! initiate a single trigger.
70 !
80     CLEAR 7                          ! Clear interface - send "device clear"
90     ASSIGN @Fgen TO 710              ! Assign I/O path to address 710
100    OUTPUT @Fgen;"*RST"              ! Reset the function generator
110 !
120    OUTPUT @Fgen;"OUTP:LOAD 50"      ! Output termination is 50 ohms
130    OUTPUT @Fgen;"APPL:SIN 5000,5"   ! Carrier is 5 kHz sine @ 5 Vpp
140    OUTPUT @Fgen;"BM:NCYC 1"         ! Burst count is 1 cycle
150    OUTPUT @Fgen;"BM:PHAS 270"       ! Starting phase is 270 degrees
160    OUTPUT @Fgen;"VOLT:OFFS 2.5"     ! Offset voltage is 2.5 Vdc
170    OUTPUT @Fgen;"TRIG:SOUR BUS"     ! Trigger source is "bus"
180    OUTPUT @Fgen;"BM:STAT ON"        ! Turn burst modulation on
190 !
200 ! Trigger the function generator over the GPIB interface
210 !
220    OUTPUT @Fgen;"*TRG"              ! or "TRIGGER 710"
230    END
```

C / GPIB (Program 1)

```
/*************************************************************************

   This program sets up a burst waveform with a 270 degree starting phase.
   By adding an offset voltage to the burst, a "haversine" is created.
   This program also shows the use of a trigger received over the GPIB
   interface to initiate a single burst.

*************************************************************************/

#include <stdio.h>          /* Used for printf() */
#include <stdlib.h>         /* Used for atoi() */
#include <string.h>         /* Used for strlen() */
#include <cfunc.h>          /* Header file from GPIB Command Library */

#define ADDR 710L           /* Set GPIB address for function generator */

/* Function Prototypes */
void rst_clear(void);
void out_waveform(void);
void command_exe(char *commands[], int length);
void burst_trig(void);
void check_error(char *func_name);

/*************************************************************************/

void main(void)             /* Start of main() */
{
  rst_clear();              /* Reset the instrument and clear error queue */
  out_waveform();           /* Set up burst modulation parameters */
  burst_trig();             /* Trigger the function generator */
}

/*************************************************************************/

void rst_clear(void)
{
  /* Reset the function generator, clear the error queue, and wait for
     commands to complete. A "1" is sent to the output buffer from the
     *OPC? command when *RST and *CLS are completed. */

  float value;
  IOOUTPUTS(ADDR, "*RST;*CLS;*OPC?", 15);
  IOENTER(ADDR, &value);
}

/*************************************************************************/
```

. . . continued

```
void out_waveform(void)
{
  /* Set up the burst waveform with a burst count of 1 and a starting
     phase of 270 degrees. The burst waveform is a 5 kHz sine wave
     with an output amplitude of 5 Vpp. An offset voltage of 2.5 Vdc
     is added to create a "haversine" wave. The "triggered burst" mode is
     used and the function generator will output a single burst when a
     "bus" trigger is received over the GPIB interface. */

  static char *cmd_string[]=
  {
    "OUTP:LOAD 50",         /* Output termination is 50 ohms */
    "APPL:SIN 5000,5",      /* Carrier is a 5 kHz sine wave @ 5 Vpp */
    "BM:NCYC 1",            /* Burst count is 1 cycle */
    "BM:PHAS 270",          /* Starting phase is 270 degrees */
    "VOLT:OFFS 2.5",        /* Offset voltage is 2.5 Vdc */
    "TRIG:SOUR BUS",        /* Trigger source is "bus" */
    "BM:STAT ON"            /* Turn burst modulation on */
  };

  /* Call the function to execute the command strings shown above */
  command_exe(cmd_string, sizeof(cmd_string)/sizeof(char*));

  /* Call the function to check for errors */
  check_error("out_waveform");
}

/*************************************************************************/

void command_exe(char *commands[], int length)
{
  /* Execute one command string at a time using a loop */

  int loop;

  for (loop = 0; loop < length; loop++)
  {
    IOOUTPUTS(ADDR, commands[loop], strlen(commands[loop]));
  }
}

/*************************************************************************/
```

. . . continued

```
void check_error(char *func_name)
{
  /* Read error queue to determine if errors have occurred */

  char message[80];
  int length = 80;

  IOOUTPUTS(ADDR, "SYST:ERR?", 9);      /* Read the error queue */
  IOENTERS(ADDR, message, &length);     /* Enter error string */

  while (atoi(message) != 0)            /* Loop until all errors are read */
  {
    printf("Error %s in function %s\n\n", message, func_name);
    IOOUTPUTS(ADDR, "SYST:ERR?", 9);
    IOENTERS(ADDR, message, &length);
  }
}

/*****************************************************************************/

void burst_trig(void)
{
  /* Trigger the function generator to output a single burst. To output
     a continuous burst change the trigger source to "IMMediate".
     To change the trigger source, send "TRIG:SOUR IMM" in the
     "out_waveform" function. */

  IOOUTPUTS(ADDR, "*TRG", 4);           /* or "IOTRIGGER(ADDR);" */
}

/*****************************************************************************/
```

End of Program 1

Using the Low-Level Commands

This program teaches the following concepts:

- How to use the low-level commands to set up an AM waveform.
- How to use the `*SAV` command to store the instrument configuration in memory.

BASIC / GPIB (Program 2)

```
10 !
20 ! This program uses low-level SCPI commands to configure
30 ! the function generator to output an AM waveform.
40 ! This program also shows how to use "state storage" to
50 ! store the instrument configuration in memory.
60 !
70     CLEAR 7                          ! Clear interface - send "device clear"
80     ASSIGN @Fgen TO 710              ! Assign I/O path to address 710
90     OUTPUT @Fgen;"*RST"              ! Reset the function generator
100 !
110    OUTPUT @Fgen;"OUTP:LOAD 50"      ! Output termination is 50 ohms
120    OUTPUT @Fgen;"FUNC:SHAP SIN"     ! Carrier waveshape is sine wave
130    OUTPUT @Fgen;"FREQ 5000;VOLT 5"  ! Carrier frequency is 5 kHz @ 5 Vpp
140    OUTPUT @Fgen;"AM:INT:FUNC SIN"   ! Modulating waveshape is sine
150    OUTPUT @Fgen;"AM:INT:FREQ 200"   ! Modulating frequency is 200 Hz
160    OUTPUT @Fgen;"AM:DEPT 80"        ! Modulation depth is 80%
170    OUTPUT @Fgen;"AM:STAT ON"        ! Turn AM modulation on
180 !
190    OUTPUT @Fgen;"*SAV 1"            ! Store state in memory location 1
200 !
210 ! Use the "*RCL 1" command to recall the stored state
220 !
230    END
```

QuickBASIC / GPIB (Program 2)

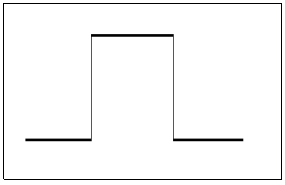
```
REM $INCLUDE: 'QBSETUP'
'
' This program uses low-level SCPI commands to configure
' the function generator to output an AM waveform.
' This program also shows how to use "state storage" to
' store the instrument configuration in memory. The program
' is written in QuickBASIC and uses the 82335B GPIB card
' and GPIB command library.
'

ISC& = 7                    ' GPIB select code is "7"
Dev& = 710                  ' Assign I/O path to address 710

Timeout = 5                 ' Configure device library for a 5 second timeout
CALL IOTIMEOUT(ISC&, Timeout)
IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR

CALL IORESET(ISC&)          ' Reset the 82335B GPIB card
IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR

CALL IOCLEAR(Dev&)          ' Send a device clear to the function generator
IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR

CALL IOREMOTE(Dev&)         ' Place the function generator in the remote mode
IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR

Info1$ = "*RST"             ' Reset the function generator
Length1% = LEN(Info1$)
CALL IOOUTPUTS(Dev&, Info1$, Length1%)
IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR

Info1$ = "OUTP:LOAD 50"     ' Output termination is 50 ohms
Length1% = LEN(Info1$)
CALL IOOUTPUTS(Dev&, Info1$, Length1%)
IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR

Info1$ = "FUNC:SHAP SIN"    ' Carrier waveshape is sine wave
Length1% = LEN(Info1$)
CALL IOOUTPUTS(Dev&, Info1$, Length1%)
IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR

Info1$ = "FREQ 5000;VOLT 5"  ' Carrier frequency is 5 kHz @ 5 Vpp
Length1% = LEN(Info1$)
CALL IOOUTPUTS(Dev&, Info1$, Length1%)
IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
```

. . . continued

```
Info1$ = "AM:INT:FUNC SIN"  ' Modulating waveshape is sine wave
Length1% = LEN(Info1$)
CALL IOOUTPUTS(Dev&, Info1$, Length1%)
IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR

Info1$ = "AM:INT:FREQ 200"  ' Modulating frequency is 200 Hz
Length1% = LEN(Info1$)
CALL IOOUTPUTS(Dev&, Info1$, Length1%)
IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR

Info1$ = "AM:DEPT 80"       ' Modulating depth is 80%
Length1% = LEN(Info1$)
CALL IOOUTPUTS(Dev&, Info1$, Length1%)
IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR

Info1$ = "AM:STAT ON"       ' Turn AM modulation on
Length1% = LEN(Info1$)
CALL IOOUTPUTS(Dev&, Info1$, Length1%)
IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR

Info1$ = "*SAV 1"           ' Store state in memory location 1
Length1% = LEN(Info1$)
CALL IOOUTPUTS(Dev&, Info1$, Length1%)
IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
'
' Use the "*RCL 1" command to recall the stored state
'

END
```

End of Program 2

Downloading an Arbitrary Waveform over GPIB



This program teaches the following concepts:

- How to define a set of waveform points over the GPIB interface and download the points to volatile memory. The downloaded waveform is a square wave pulse (4,000 points) with a calculated rise and fall time.
- How to copy the download waveform to non-volatile memory.

BASIC / GPIB (Program 3)

```
10 !
20 ! This program uses the arbitrary waveform function to
30 ! download and output a square wave pulse with a calculated
40 ! rise time and fall time. The waveform consists of 4000
50 ! points downloaded to the function generator as ASCII data.
60 !
70     OPTION BASE 1
80     DIM Waveform(4000)       ! Dimension array with 4000 elements
90     INTEGER I                ! Declare integer variable
100    CLEAR 7                  ! Clear interface - send "device clear"
110    ASSIGN @Fgen TO 710      ! Assign I/O path to address 710
120    OUTPUT @Fgen;"*RST"      ! Reset the function generator
130 !
140    FOR I=1 TO 5
150      Waveform(I)=(I-1)/5    ! Set rise time (5 points)
160    NEXT I
170 !
180    FOR I=6 TO 205
190      Waveform(I)=1          ! Set pulse width (200 points)
200    NEXT I
210 !
220    FOR I=206 TO 210
230      Waveform(I)=(210-I)/5  ! Set fall time (5 points)
240    NEXT I
250 !
260    FOR I=211 TO 4000
270      Waveform(I)=0          ! Set remaining points to zero
280    NEXT I
290 !
```

. . . continued

```
300 ! Download data points to volatile memory from array
310 !
320    DISP "Downloading Arb..."
330    OUTPUT @Fgen;"DATA VOLATILE,";Waveform(*)
340    DISP "Download Complete"
350 !
360    OUTPUT @Fgen;"DATA:COPY PULSE, VOLATILE"  ! Copy arb to non-volatile memory
370    OUTPUT @Fgen;"FUNC:USER PULSE"            ! Select the active arb waveform
380    OUTPUT @Fgen;"FUNC:SHAP USER"             ! Output the selected arb waveform
390 !
400    OUTPUT @Fgen;"OUTP:LOAD 50"               ! Output termination is 50 ohms
410    OUTPUT @Fgen;"FREQ 5000;VOLT 5"           ! Output frequency is 5 kHz @ 5 Vpp
420 !
430    END
```

C / GPIB (Program 3)

```
/************************************************************************

   This program uses the arbitrary waveform function to download and
   output a square wave pulse with a calculated rise time and fall time.
   The waveform consists of 4000 points downloaded over the GPIB
   interface as ASCII data.

************************************************************************/

#include <stdio.h>         /* Used for printf() */
#include <stdlib.h>        /* Used for malloc(), free(), atoi() */
#include <string.h>        /* Used for strlen() */
#include <cfunc.h>         /* Header file from GPIB Command Library */

#define ISC 7L             /* Assign GPIB select code */
#define ADDR 710L          /* Set GPIB address for function generator */

/* Function Prototypes */
void rst_clear(void);
void get_data(void);
void download_data(float *waveform, int num_points);
void out_waveform(void);
void command_exe(char *commands[], int length);
void check_error(char *func_name);

/************************************************************************/
```

. . . continued

```
void main(void)            /* Start of main() */
{
  rst_clear();             /* Reset the instrument and clear error queue */
  get_data();              /* Calculate the waveform data points */
  out_waveform();          /* Download points and output arb waveform */
}

/*************************************************************************/

void rst_clear(void)
{
  /* Reset the function generator, clear the error queue, and wait for
     commands to complete. A "1" is sent to the output buffer from the
     *OPC? command when *RST and *CLS are completed. */

  float value;

  IOOUTPUTS(ADDR, "*RST;*CLS;*OPC?", 15);
  IOENTER(ADDR, &value);
}

/*************************************************************************/

void get_data(void)
{
  /* Load 4000 points into an array to set the rise time and fall time
     to 250 ns and the pulse width to 10 us (the output frequency is set
     to 5 kHz in the "out_waveform" function). */

  float *waveform;
  int loop, num_points = 4000;
  waveform = (float*) malloc (num_points * sizeof(float));

  for (loop = 1; loop <= 5; loop++)
  {
    waveform[loop] = (float)(loop-1)/5;   /* Set rise time (5 points) */
  }

  for (loop = 6; loop <= 205; loop++)
  {
    waveform[loop] = 1;                   /* Set pulse width (10 points) */
  }

  for (loop = 206; loop <= 210; loop++)
  {
    waveform[loop] = (float)(210-loop)/5; /* Set fall time (5 points) */
  }
```

. . . continued

```
  for (loop = 211; loop <= 4000; loop++)
  {
    waveform[loop] = 0;                    /* Set remaining points to zero */
  }

  /* Call function to download the 4000 data points to volatile memory */
  download_data(waveform, num_points);

  /* Free memory allocated for data points */
  free(waveform);
}

/*****************************************************************************/

void download_data(float *waveform, int num_points)
{
  /* Download the waveform array (4000 points) to volatile memory.
     The function generator expects to receive the arb waveform data as
     one contiguous block. To do this, suppress the carriage return (CR)
     and line feed (LF) before downloading the data. */

  static char state[2] = {13, 10};     /* ASCII 13 = carriage return
                                          ASCII 10 = line feed */

  /* First, disable EOI (End-or-Identify) and EOL (End-of-Line) */
  IOEOI (ISC,0); IOEOL (ISC, " ", 0);

  /* Send "DATA VOLATILE" header and suppress CR/LF */
  printf("Downloading Arb...\n\n");
  IOOUTPUTS (ADDR, "DATA VOLATILE,", 14);

  /* Re-enable EOI and EOL for normal GPIB operation and send the data */
  IOEOI (ISC,1); IOEOL (ISC, state, 2);

  /* Use the IOOUTPUTA to output the waveform data as an ASCII array */
  IOOUTPUTA (ADDR, waveform, num_points);  printf("Download Complete\n\n");

  /* Call the function to check for errors */
  check_error("download_data");
}

/*****************************************************************************/
```

. . . continued

```
void out_waveform(void)
{
  /* Set the output frequency to 5 kHz with an amplitude of 10 Vpp and
     output the arbitrary waveform. */

  static char *cmd_string[]=
  {
    /* Copy arb waveform to non-volatile memory with name "PULSE" */
    "DATA:COPY PULSE, VOLATILE",
    "FUNC:USER PULSE",      /* Select the active arb waveform */
    "FUNC:SHAP USER"        /* Output the selected arb waveform */
  };

  /* Call the function to execute the command strings shown above */
  command_exe(cmd_string, sizeof(cmd_string)/sizeof(char*));

  /* Call the function to check for errors */
  check_error("out_waveform");
}

/*****************************************************************************/

void command_exe(char *commands[], int length)
{
  /* Execute one command string at a time using loop */

  int loop;

  for (loop = 0; loop < length; loop++)
  {
    IOOUTPUTS(ADDR, commands[loop], strlen(commands[loop]));
  }

  /* Set output termination to 50 ohms and output frequency to 5 kHz @ 5 Vpp */
  IOOUTPUTS(ADDR, "OUTP:LOAD 50", 12);
  IOOUTPUTS(ADDR, "FREQ 5000;VOLT 5", 16);
}

/*****************************************************************************/
```

. . . continued

```
void check_error(char *func_name)
{
  /* Read error queue to determine if errors have occurred */

  char message[80];
  int length = 80;

  IOOUTPUTS(ADDR, "SYST:ERR?", 9);      /* Read the error queue */
  IOENTERS(ADDR, message, &length);     /* Enter error string */

  while (atoi(message) != 0)            /* Loop until all errors are read */
  {
    printf("Error %s in function %s\n\n", message, func_name);
    IOOUTPUTS(ADDR, "SYST:ERR?", 9);
    IOENTERS(ADDR, message, &length);
  }
}

/*************************************************************************/
```

End of Program 3

Using the Status Registers

This program teaches the following concepts:

- How to use the Status Registers to generate an interrupt if a SCPI error occurs. The program sets up the Status Byte and Standard Event register and interrupts the controller if an error is detected.
- How to service the interrupt if an error occurs and read the function generator's error queue using the `SYST:ERR?` command.

BASIC / GPIB (Program 4)

```
10 !
20 ! This program uses the status registers to generate an
30 ! interrupt if a SCPI error occurs. The function generator
40 ! is programmed to output an FM waveform.
50 !
60     CLEAR 7                      ! Clear interface - send "device clear"
70     ASSIGN @Fgen TO 710          ! Assign I/O path to address 710
80     COM @Fgen                    ! Use same address in subprogram
90     INTEGER Hpib,Mask,Value,B    ! Declare integer variables
100    OUTPUT @Fgen;"*RST"          ! Reset function generator
110 !
120 ! Set up error checking
130 !
140    Hpib=7                       ! GPIB select code is "7"
150    ON INTR Hpib CALL Err_msg    ! Call subprogram if error occurs
160    Mask=2                       ! Bit 1 is SRQ
170    ENABLE INTR Hpib;Mask        ! Enable SRQ to interrupt program
180 !
190    OUTPUT @Fgen;"*CLS"          ! Clear status registers (clear out old errors)
200    OUTPUT @Fgen;"*SRE 32"       ! Enable "Standard Event" bit in Status Byte
210                                 ! to pull the IEEE-488 SRQ line
220    OUTPUT @Fgen;"*ESE 60"       ! Enable error bits (2, 3, 4, or 5) to set
230                                 ! "Standard Event" bit in Status Byte
240                                 ! and wait for operation complete
```

. . . continued

```
250 !
260 ! Set up function generator to output FM waveform
270 !
280    OUTPUT @Fgen;"OUTP:LOAD 50"       ! Output termination is 50 ohms
290    OUTPUT @Fgen;"APPL:SIN 5000,5"    ! Carrier waveshape is sine function,
300                                      ! output frequency is 5 kHz @ 5 Vpp
310    OUTPUT @Fgen;"FM:INT:FUNC SIN"    ! Modulating waveshape is sine wave
320    OUTPUT @Fgen;"FM:INT:FREQ 500"    ! Modulating frequency is 500 Hz
330    OUTPUT @Fgen;"FM:DEV 250"         ! Frequency deviation is  250 Hz
340    OUTPUT @Fgen;"FM:STAT ON"         ! Turn FM modulation on
350 !
360    OUTPUT @Fgen;"*OPC?"              ! Send "1" to output buffer when complete
370    ENTER @Fgen;Value                 ! Enter value
380 !
390    OFF INTR Hpib                     ! Disable interrupts
400    END
410 !
420    SUB Err_msg  ! Error subprogram is called if errors occurred
430      DIM Message$[80]                ! Dimension array for error
440      INTEGER Code                    ! Define integer variable
450      COM @Fgen                       ! Use same address as in main program
460      B=SPOLL(@Fgen)                  ! Use Serial Poll to read Status Byte
470                                      ! (all bits are cleared too)
480 !
490 ! Loop until error queue is cleared
500 !
510      REPEAT
520        OUTPUT @Fgen;"SYST:ERR?"
530        ENTER @Fgen;Code,Message$
540        PRINT Code,Message$
550      UNTIL Code=0
560      STOP
570    SUBEND
```

C / GPIB (Program 4)

```
/**************************************************************************

   This program uses the Agilent 33120A Status Registers to generate an
   interrupt if a SCPI error is detected. The function generator is
   programmed to output an FM waveform.

**************************************************************************/

#include <stdio.h>          /* Used for printf() */
#include <stdlib.h>         /* Used for atoi() */
#include <string.h>         /* Used for strlen() */
#include <cfunc.h>          /* Header file from GPIB Command Library */

#define ISC 7L              /* Assign GPIB select code */
#define ADDR 710L           /* Set GPIB address for function generator */

/* Function Prototypes */
void rst_clear(void);
void setup_status(void);
void out_waveform(void);
void command_exe(char *commands[], int length);
void check_error(char *func_name);

/**************************************************************************/

void main(void)             /* Start of main() */
{
  rst_clear();              /* Reset the instrument and clear error queue */
  setup_status();           /* Set up 33120A status registers */
  out_waveform();           /* Set up frequency sweep parameters */
}

/**************************************************************************/
```

. . . continued

```
void rst_clear(void)
{
  /* Reset the function generator, clear the error queue, and wait for
     commands to complete. A "1" is sent to the output buffer from the
     *OPC? command when *RST and *CLS are completed. */

  float value;

  IOOUTPUTS(ADDR, "*RST;*CLS;*OPC?", 15);
  IOENTER(ADDR, &value);
}

/*************************************************************************/

void setup_status(void)
{
  /* An interrupt is to be generated whenever an error is detected.
     To enable the status registers to generate an interrupt, you must
     enable bits in the Status Byte and Standard Event Register. */

  /* Enable bit 5 "Standard Event" in the Status Byte to recognize
     activity from the Standard Event Register (2^5 = 32) */

  IOOUTPUTS(ADDR, "*SRE 32", 7);

  /* Enable bits 2, 3, 4, and 5 in the Standard Event Register to
     interrupt the Status Byte (2^2 + 2^3 + 2^4 + 2^5 = 60) */

  IOOUTPUTS(ADDR, "*ESE 60", 7);
}

/*************************************************************************/
```

. . . continued

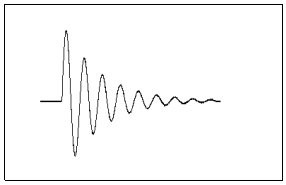
```
void out_waveform(void)
{
  /* Configure the waveform and enable FM modulation. The carrier waveform
     is a 5 kHz sine wave and the modulating waveform is a 500 Hz sine wave.
     The peak frequency deviation is 250 Hz. */

  float value;
  int stat_cond;

  static char *cmd_string[]=
  {
    "OUTP:LOAD 50",         /* Output termination is 50 ohms */
    "APPL:SIN 5000,5",      /* Carrier waveshape is sine wave,
                               output frequency is 5 kHz @ 5 Vpp */
    "FM:INT:FUNC SIN",      /* Modulating waveshape is sine wave */
    "FM:INT:FREQ 500",      /* Modulating frequency is 500 Hz */
    "FM:DEV 250"            /* Frequency deviation is 250 Hz */
  };

  /* Call the function to execute the command strings shown above */
  command_exe(cmd_string, sizeof(cmd_string)/sizeof(char*));

  /* Turn sweep on and send "1" to output buffer when setup is complete */
  IOOUTPUTS(ADDR, "FM:STAT ON;*OPC?", 17);
  IOENTER(ADDR, &value);

  /* Determine the status of the SRQ line; return "1" if error occurred */
  IOSTATUS(ISC, 1, &stat_cond);

  if (stat_cond)
  {
    check_error("out_waveform");  /* Call function to check for errors */
  }
  else
  {
    printf("No Errors Were Detected.\n\n");
  }

  /* Use Serial Poll to read the Status Byte and clear all bits */
  IOSPOLL(ADDR, &stat_cond);
}

/*************************************************************************/
```

. . . continued

```
void command_exe(char *commands[], int length)
{
  /* Execute one command string at a time using a loop */

  int loop;

  for (loop = 0; loop < length; loop++)
  {
    IOOUTPUTS(ADDR, commands[loop], strlen(commands[loop]));
  }
}

/*************************************************************************/

void check_error(char *func_name)
{
  /* Read error queue to determine if errors have occurred */

  char message[80];
  int length = 80;

  IOOUTPUTS(ADDR, "SYST:ERR?", 9);      /* Read the error queue */
  IOENTERS(ADDR, message, &length);     /* Enter error string */

  while (atoi(message) != 0)            /* Loop until all errors are read */
  {
    printf("Error %s in function %s\n\n", message, func_name);
    IOOUTPUTS(ADDR, "SYST:ERR?", 9);
    IOENTERS(ADDR, message, &length);
  }
}

/*************************************************************************/
```

End of Program 4

Downloading an Arbitrary Waveform over RS-232



This program teaches the following concepts:

- How to define a set of binary waveform points and download the points to volatile memory. The downloaded waveform is a damped sine wave.
- How to copy the download waveform to non-volatile memory.

BASIC / RS-232 (Program 5)

```
10 !
20 ! This program shows how to download an arbitrary waveform
30 ! using binary data over the RS-232 interface. The program
40 ! generates a damped sine wave using 16,000 points.
50 !
60     OPTION BASE 1
70     CONTROL 10,3;9600                  ! Configure PC for 9600 baud (COM2)
80     CONTROL 10,4;7                     ! Configure PC for 8 bits with no parity
90     INTEGER Waveform(16000),I,Ncycles  ! Dimension array and variables
100    REAL Damp_factor                   ! Define real variable
110    ASSIGN @Fgen TO 10                 ! Device address is 10
120    ASSIGN @Bin TO 10;FORMAT OFF       ! Prepare interface for binary data
130    OUTPUT @Fgen;"*RST"                ! Reset the function generator
140 !
150    OUTPUT @Fgen;"SYST:REM"            ! Enable the remote RS-232 mode
160    OUTPUT @Fgen;"FORM:BORD SWAP"      ! Swap data bytes (send LSB first)
170    OUTPUT @Fgen;"FREQ 5000"           ! Output frequency is 5 kHz
180    OUTPUT @Fgen;"OUTP:LOAD 50"        ! Output termination is 50 ohms
190    OUTPUT @Fgen;"VOLT 5"              ! Output amplitude is 5 Vpp
200 !
210    Ncycles = 10                       ! Define number of cycles
220    Damp_factor = -5                   ! Define damping factor
230 !
240 ! Calculate data points
250 !
260    DISP "Calculating Data Points..."
270    FOR I=1 TO 16000
280      IF I<14000 THEN
290        Waveform(I)=SIN(2*PI*Ncycles*I/14000)*2047
300        Waveform(I)=Waveform(I)*EXP(Damp_factor*I/14000)
310      ELSE
320        Waveform(I)=0
330      END IF
340    NEXT I
```

. . . continued

```
350 !
360 ! Download data points to volatile memory
370 !
380    DISP "Downloading Arb..."
390    OUTPUT @Fgen;"DATA:DAC VOLATILE,#532000";     ! Output binary block header
400    WAIT 1 ! Wait for interface
410    OUTPUT @Bin;Waveform(*);                      ! Output binary block
420    WAIT 1 ! Wait for interface
430    OUTPUT @Fgen;";*OPC?"    ! Send terminator and wait for download to complete
440    ENTER @Fgen;Value ! Enter value
450    DISP "Download Complete"
460 !
470    OUTPUT @Fgen;"DATA:COPY DAMP_SINE, VOLATILE"  ! Copy to non-volatile memory
480    OUTPUT @Fgen;"FUNC:USER DAMP_SINE"            ! Select the active waveform
490    OUTPUT @Fgen;"FUNC:SHAP USER"                 ! Output the selected waveform
500    END
```

QuickBASIC / RS-232 (Program 5)

```
' This program shows how to download an arbitrary waveform
' using ASCII data over the RS-232 interface. The program
' generates a damped sine wave using 1,000 points.
'
npnts = 1000          ' Define number of ASCII points in waveform
NCYCLES = 10          ' Define number of cycles
DAMPFACTOR = -5       ' Define damping factor

DIM waveform(npnts)   ' Dimension waveform array

'
' Configure COM2 for 9600 baud, even parity, 7 data bits, 2 stop bits,
' suppress detection of Request to Send (rs), set timeout of Data Carrier
' Detect line (cd), and terminate output with line feed (lf).
'
OPEN "com2:9600,e,7,2,rs,cd,lf,pe" FOR RANDOM AS #1 LEN = 1000
'
PRINT #1, ":SYST:REM"        ' Enable the remote RS-232 mode
'
PRINT #1, "*RST"             ' Reset the function generator
PRINT #1, "FORM:BORD SWAP"   ' Swap data bytes (send LSB first)
PRINT #1, "FREQ 5000"        ' Output frequency is 5 kHz
PRINT #1, "OUTP:LOAD 50"     ' Output termination is 50 ohms
PRINT #1, "VOLT 5"           ' Output amplitude is 5 Vpp
```

. . . continued

```
'
' Calculate data points
'
PRINT "Calculating Data Points..."
pi = 3.1415
FOR I = 1 TO npnts
  waveform(I) = EXP(DAMPFACTOR * I / npnts) * SIN(2 * pi * NCYCLES * I / npnts)
NEXT I
'
' Download data points to volatile memory
'
PRINT "Downloading Arb..."
PRINT #1, "DATA VOLATILE,";
FOR I = 1 TO npnts - 1
  PRINT #1, STR$(waveform(I)) + ",";
NEXT I
PRINT #1, STR$(waveform(npnts))
'
PRINT #1, "*OPC?"    ' Wait for download to complete
LINE INPUT #1, resp$

PRINT "Download Complete"
PRINT #1, "DATA:COPY DAMP_SIN, VOLATILE"  ' Copy to non-volatile memory
PRINT #1, "FUNC:USER DAMP_SIN"            ' Select the active arb
PRINT #1, "FUNC:SHAP USER"                ' Output the selected arb
PRINT #1, "*OPC?" LINE INPUT #1, resp$
PRINT "Program Complete"

END
```

End of Program 5

Tutorial

Tutorial

The Agilent 33120A is capable of producing a variety of signal waveshapes. In order to achieve the greatest performance from the function generator, it may be helpful if you learn more about the internal operations of the instrument. This chapter describes basic signal generation concepts and gives specific details on the internal operations of the function generator.

- Direct Digital Synthesis, *starting on page 273*
- Signal Imperfections, *starting on page 276*
- Creating Arbitrary Waveforms, *starting on page 278*
- Output Amplitude Control, *starting on page 280*
- Floating Signal Generators, *on page 282*
- Attributes of AC Signals, *starting on page 283*
- Modulation, *starting on page 287*

You can use an arbitrary waveform generator in a variety of applications where it might be otherwise difficult or impossible to generate complex output waveforms. With an arbitrary waveform generator, signal imperfections such as rise time, ringing, glitches, noise, and random timing variations can be easily simulated in a controlled manner.

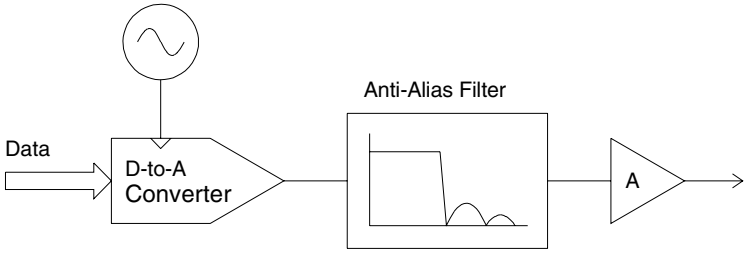
Physics, chemistry, biomedicine, electronics, mechanics, and other fields can benefit from the versatility of an arbitrary waveform generator. Wherever things vibrate, pump, pulse, bubble, burst, or change with time in any way, there are possible applications — limited only by your ability to provide waveform data.

The *Agilent IntuiLink Arb Waveform Generation Software* for Microsoft® Windows® is designed to make it easy to create and output arbitrary waveforms for the 33120A.

Direct Digital Synthesis

Digital signal processing methods are used in many everyday applications. Whether it is a digital audio compact disc player, an electronic synthesized piano, or a voice-synthesized telephone message system, it is obvious that complex waveforms can be easily created or reproduced using digital signal generation methods.

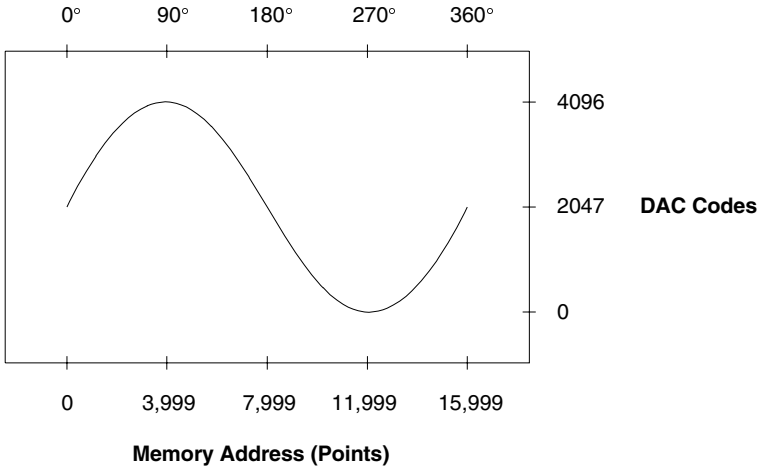
The 33120A uses a signal-generation technique called *direct digital synthesis* or DDS. The basic principle behind DDS is not unlike an audio compact disc. As shown below for digital audio, a stream of digital data representing the sampled analog signal shape is sequentially addressed from a disc. This data is applied to the digital port of a digital-to-analog converter (DAC) which is clocked at a constant rate. The digital data is then converted into a series of voltage steps approximating the original analog signal shape. After filtering the voltage steps, the original analog waveshape will be recovered. The incoming data can be of any arbitrary shape, as long as it matches the requirements of the particular DAC (16 bits for digital audio players).



Direct Digital Synthesis
(continued)

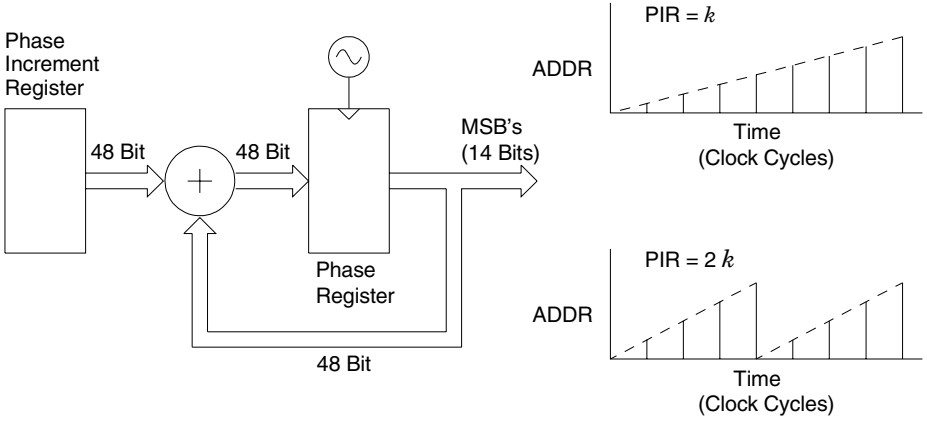
A direct digital synthesis (*DDS*) signal generator differs from a digital audio player because of its very precise control of the data stream input to the DAC. In a DDS system, the amplitude values for one complete cycle of the output waveshape are stored sequentially in random access memory (*RAM*) as shown in the figure below. As RAM addresses are changed, the DAC converts the waveshape data into a voltage waveform (whose data values are loaded in RAM). The frequency of the voltage waveform is proportional to the rate at which the RAM addresses are changed.

The 33120A represents amplitude values by 4,096 discrete voltage levels (or 12-bit vertical resolution). Waveforms may contain between 8 points and 16,000 points of 12-bit amplitude values. The number of points in RAM representing one complete cycle of the waveshape (or 360°) is called its *length* or *horizontal resolution*. Each RAM address corresponds to a phase increment equal to 360°/ *points*, where *points* is the waveform length. Therefore, sequential RAM addresses contain the amplitude values for the individual points (0° to 360°) of the waveform.



Direct digital synthesis (*DDS*) generators use a *phase accumulation* technique to control waveform RAM addressing. Instead of using a counter to generate sequential RAM addresses, an "adder" is used. On each clock cycle, the constant loaded into the phase increment register (*PIR*) is added to the present result in the phase accumulator (see below). The most-significant bits of the phase accumulator output are used to address waveform RAM — the upper 14 bits (2^{14} = 16,384 RAM addresses) for the 33120A. By changing the PIR constant, the number of clock cycles required to step through the entire waveform RAM changes, thus changing the output frequency. When a new PIR constant is loaded into the register, the waveform output frequency changes phase continuously following the next clock cycle.

The 33120A uses a 48-bit phase accumulator which yields $F_{clk}/2^{48}$ or approximately 142 nHz frequency resolution internally. The phase accumulator output (the upper 14 bits) will step sequentially through each RAM address for smaller PIR values (lower frequencies). However, when the PIR is loaded with a larger value, the phase accumulator output will skip some RAM addresses, automatically "sampling" the data stored in RAM. Therefore, as the output frequency is increased, the number of output samples per waveshape cycle will decrease. In fact, different groups of points may be output on successive waveform cycles.



The maximum output frequency, with the condition that every waveshape point in RAM is output every waveform cycle, is defined by:

$$F_{out} = F_{clk} / Points$$

The minimum number of points required to accurately reproduce a waveshape will determine the maximum useful output frequency using the same equation.

> *The rule governing waveforms is referred to as the **Nyquist Sampling Theorem**, which states that you must include at least two points from the highest frequency component of the signal you are attempting to reproduce.*

Signal Imperfections

Most signal imperfections are easiest to observe in the frequency domain using a spectrum analyzer. Sampling theory predicts the location and size of spurious signals resulting from the sampling processes used by DDS generators. In fact, since DDS generators use a fixed sampling rate (40 MHz for the 33120A), spurious signals can be removed with a fixed frequency "anti-alias" filter. A 17 MHz, ninth-order elliptical filter providing a sharp cut-off (in excess of 60 dB attenuation for signals greater than 19 MHz) is used for sine wave outputs. A 10 MHz, seventh-order Bessel filter is used for non-sine wave outputs. The Bessel filter provides slower amplitude roll-off for anti-alias filtering, but maintains linear phase response to minimize shape distortion for complex waveshapes. The 33120A automatically selects the appropriate filter when the output function is selected.

All digital-to-analog converters, including those used in DDS generators, produce spurious signals resulting from non-ideal performance. These spurious signals are harmonically related to the desired output signal. At lower frequencies, the 33120A's 12-bit waveform DAC produces spurious signals near the -74 dBc level (decibels below the carrier or output signal) as described by the equation on the following page. The 33120A uses the complete vertical resolution ($N=1$) of the DAC for all internal waveshapes, thus minimizing amplitude quantization error.

At higher output frequencies, additional DAC errors produce non-harmonic spurious outputs. These are signals "folded back" or aliased to a frequency within the signal bandwidth. A "perfect" DAC will also produce a wideband noise floor due to amplitude quantization. The noise floor for a 12-bit DAC will be near the -74 dBc level; this corresponds to a noise density of -148 dBc/Hz for sine wave outputs from the 33120A.

$$\text{Amplitude Quantization} \leq -(20 \times \log_{10}(N \times 4096) + 1.8) \quad dBc$$

where "N" is the fraction of available DAC codes used to describe the signal waveshape ($0 \leq N \leq 1$).

Another type of waveform error visible in the frequency domain is *phase truncation error*. This error results from time quantization of the output waveform. Whenever a waveshape is described by a finite number of horizontal points (length), it has been sampled in time (or quantized) causing a phase truncation error. Spurious signals caused by phase truncation introduce jitter into the output waveform. This may be regarded as time (and phase) displacement of output zero crossings.

Phase truncation causes phase modulation of the output signal which results in spurious harmonics (see the equation below). For lower output frequencies, the phase accumulator periodically *does not* advance RAM addresses, causing the DAC to deliver the same voltage as recorded on the previous clock cycle. Therefore, the phase "slips" back by 360°/ *points* before continuing to move forward again. When RAM address increments are the same on each cycle of the output, phase truncation error (and jitter) are essentially zero. All standard waveshapes in the 33120A are generated with at least 16,000 waveform points which results in spurious signals below the wide-band noise floor of the DAC.

$$\text{Phase Truncation Harmonics} \leq -20 \times \log_{10}(P) \quad dBc$$

where "P" is the number of waveform points in RAM.

Creating Arbitrary Waveforms

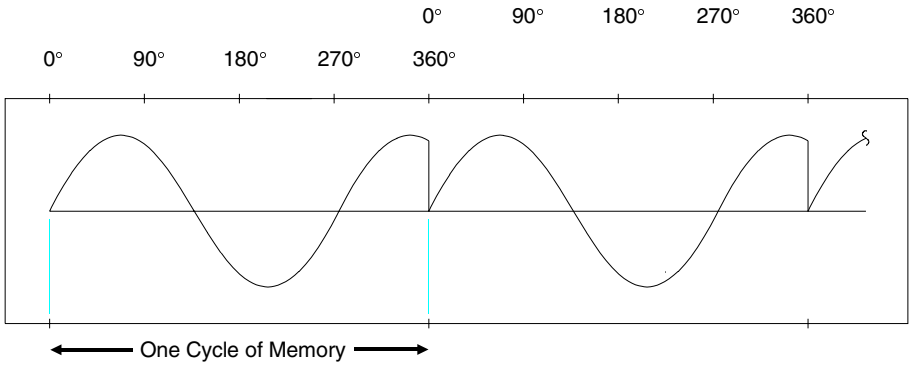
For most applications, it is not necessary to create a waveform of any specific length since the function generator will automatically sample the available data to produce an output signal. In fact, it is generally best to create arbitrary waveforms which use all available data (16,000 points long and the full range from 0 to 4,095 DAC codes). For the 33120A, you do not have to change the length of the waveform to change its output frequency. All you have to do is create a waveform of any length and then adjust the function generator's output frequency. Remember, if you create an arbitrary waveform that includes three cycles of a waveshape (for example), the output frequency will be three times the value displayed on the function generator's front panel.

When creating arbitrary waveforms, *you* have control of both the amplitude quantization and phase truncation errors. For example, phase truncation harmonics will be generated when a waveform is created using the full amplitude range of the DAC (12 bits) but is created using only 1,000 waveform data points. In this case, the amplitude quantization errors will be near the noise floor while the time quantization error will produce harmonics near the -60 dBc level. Similarly, amplitude quantization harmonics will be generated when you create a waveform using less than the full amplitude resolution of the function generator. For example, if you use only one-fifth of the available amplitude resolution, amplitude quantization will produce harmonics below the -60 dBc level.

> *When importing data from instruments such as oscilloscopes, the data will generally range between 1,024 and 4,096 time points and between 64 and 256 amplitude points.*

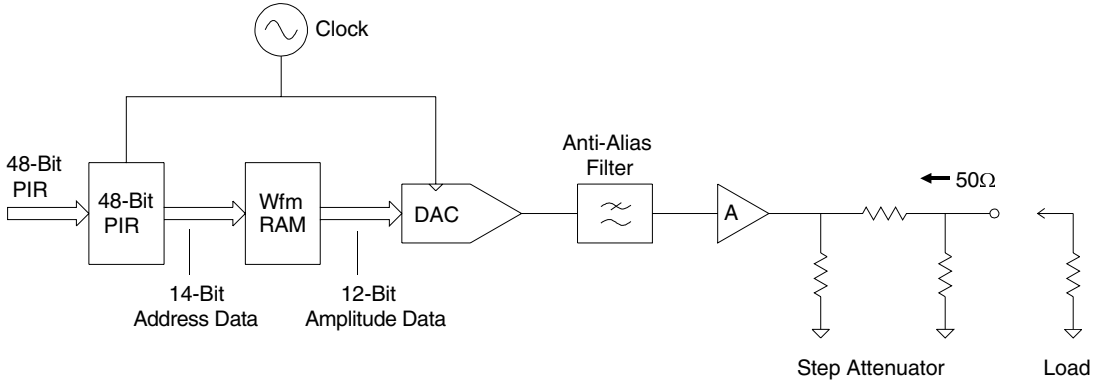
When creating arbitrary waveforms, the function generator will always attempt to replicate the finite-length time record to produce a periodic version of the data in waveform memory. As shown on the next page, it is possible that the shape and phase of a signal may be such that a transient is introduced at the end point. When the waveshape is repeated for all time, this end-point transient will introduce *leakage error* in the frequency domain because many spectral terms are required to describe the discontinuity.

Leakage error is caused when the waveform record does not include an integer number of cycles of the fundamental frequency. Power from the fundamental frequency, and its harmonics, is transferred to spectral components of the rectangular sampling function. Instead of the expected narrow spectral lines, leakage can cause significant spreading around the desired spectral peaks. You can reduce leakage errors by adjusting the window length to include an integer number of cycles or by including more cycles within the window to reduce the residual end-point transient size. Some signals are composed of discrete, non-harmonically related frequencies. Since these signals are non-repetitive, all frequency components cannot be harmonically related to the window length. You should be careful in these situations to minimize end-point discontinuities and spectral leakage.

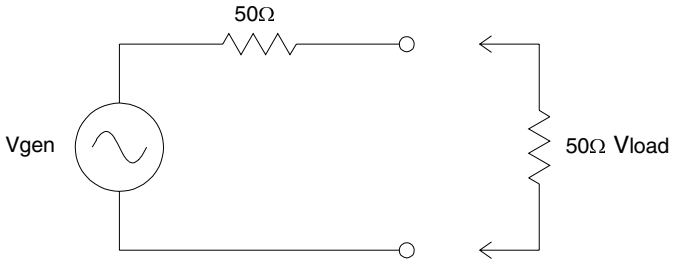


Output Amplitude Control

The 33120A uses a 12-bit digital-to-analog converter (*DAC*) to convert the digital representation of a signal to an analog output voltage. The DAC can create waveforms represented by 4,096 (2^{12}) discrete amplitude levels. All 4,096 amplitude codes are used for the built-in waveforms. Output levels from full maximum to minimum output are controlled by applying varying amounts of signal gain or attenuation to the signal created by the DAC as shown in the block diagram below. The output waveform is always described by the full 12-bit vertical resolution. You can download user-defined arbitrary waveforms using less than the full 12-bit vertical resolution; however, it is recommended that you always use the full DAC amplitude resolution to minimize amplitude quantization errors as previously discussed.



As shown below, the 33120A has a fixed output source resistance of 50 ohms. During calibration, output amplitudes are calibrated for both the open-circuit voltage (no load) and the terminated output voltage (loaded). The terminated output amplitude is calibrated for an exact 50 ohm load. Since the function generator's output resistance and the load resistance form a voltage divider, the measured output voltage of the function generator will vary with load resistance value and accuracy. When the function generator's output is loaded with a 0.2% accuracy termination, an additional (negligible) 0.2% amplitude error is created. Using a 5% accuracy termination will add 5% additional error to specified output amplitudes.

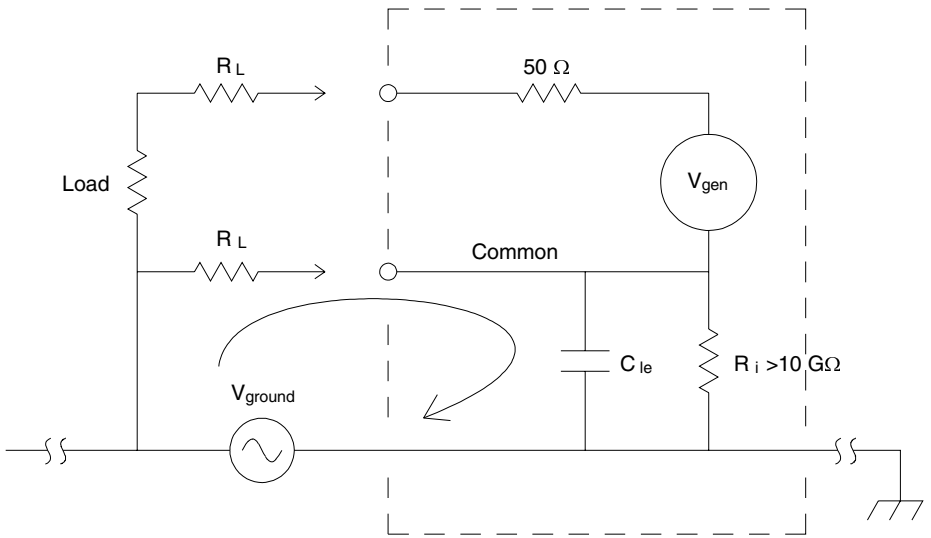


If the function generator's output is measured with no load connected, the output will be approximately *twice* the displayed amplitude (V*gen* instead of V*load)*. In some applications, you might continually use the function generator in a "no-load" conditions. In such applications, remembering to *double* the function generator's displayed amplitude can cause many errors. The 33120A allows you to specify the function generator's load condition using the `OUTPUT:LOAD` command; thus enabling the function generator to display the correct output amplitude.

Floating Signal Generators

Many applications require a test signal which is isolated from earth ground for connection to powered circuits, to avoid ground loops, or to minimize other common mode noise. A floating signal generator such as the 33120A has both sides of the output BNC connector isolated from chassis (earth) ground. As shown in the figure below, any voltage difference between the two ground reference points (V*ground*) causes a current to flow through the function generator's output common lead. This can cause errors such as noise and offset voltage (usually power-line frequency related), which are added to the output voltage.

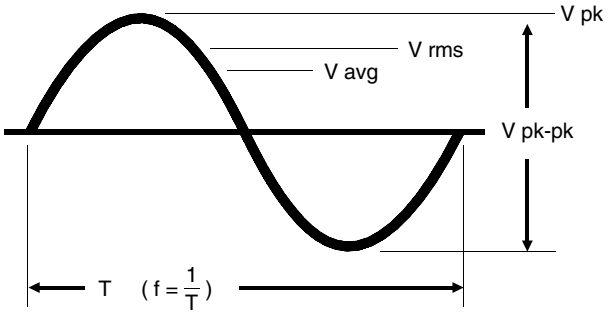
The best way to eliminate ground loops is to maintain the function generator's isolation from earth ground. The function generator's isolation impedance will be reduced as the frequency of the V*ground* source increases due to low-to-earth capacitance C*le* (approximately 4000 pF for the 33120A). If the function generator must be earth-referenced, be sure to connect it (and the load) to the same common ground point. This will reduce or eliminate the voltage difference between devices. Also, make sure the function generator and load are connected to the same electrical outlet if possible.



R L = Lead Resistance

Attributes of AC Signals

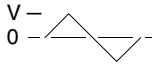
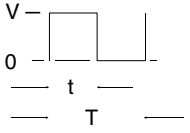
The most common ac signal is the sine wave. In fact, all periodic waveshapes are composed of sine waves of varying frequency, amplitude, and phase added together. The individual sine waves are harmonically related to each other — that is to say, the sine wave frequencies are integer multiples of the lowest (or fundamental) frequency of the waveform. Unlike dc signals, the amplitude of ac waveforms varies with time as shown in the following figure.



A sine wave can be uniquely described by any of the parameters indicated — the peak-to-peak value, peak value, or RMS value, and its period (T) or frequency (1/T).

AC Attributes
(continued)

The magnitude of a sine wave can be described by the RMS value (effective heating value), the peak-to-peak value (2 x zero-to-peak), or the average value. Each value conveys information about the sine wave. The table below shows several common waveforms with their respective peak and RMS values.

Waveform Shape	Crest Factor (C.F.)	AC RMS	AC+DC RMS
	1.414	$\frac{V}{1.414}$	$\frac{V}{1.414}$
	1.732	$\frac{V}{1.732}$	$\frac{V}{1.732}$
	$\sqrt{\frac{T}{t}}$	$\frac{V}{C.F.} \times \sqrt{1 - \left(\frac{1}{C.F.}\right)^2}$	$\frac{V}{C.F.}$

Each waveshape exhibits a zero-to-peak value of "V" volts.
Crest factor refers to the ratio of the peak-to-RMS value of the waveform.

RMS The RMS value is the only measured amplitude characteristic of a waveform that does not depend on waveshape. Therefore, the RMS value is the most useful way to specify ac signal amplitudes. The RMS value (or equivalent heating value) specifies the ability of the ac signal to deliver power to a resistive load (heat). The RMS value is equal to the dc value which produces the same amount of heat as the ac waveform when connected to the same resistive load.

For a dc voltage, this heat is directly proportional to the amount of power dissipated in the resistance. For an ac voltage, the heat in a resistive load is proportional to the average of the instantaneous power dissipated in the resistance. This has meaning only for periodic signals. The RMS value of a periodic waveform can be obtained by taking the dc values at each point along one complete cycle, squaring the values at each point, finding the average value of the squared terms, and taking the square-root of the average value. This method leads to the RMS value of the waveform regardless of the signal shape.

Peak-to-Peak and Peak Value The zero-to-peak value is the maximum positive voltage of a waveform. Likewise, the peak-to-peak value is the magnitude of the voltage from the maximum positive voltage to the most negative voltage peak. The peak or peak-to-peak amplitude of a complex ac waveform does not necessarily correlate to the RMS heating value of the signal. When the specific waveform is known, you can apply a correction factor to convert peak or peak-to-peak values to the correct RMS value for the waveform.

Average Value The average value of an ac waveform is the average of the instantaneous values measured over one complete cycle. For sine waves, the average amplitude is zero since the waveform has equal positive and negative half cycles. Since the quantity of interest is the heating value of the signal, the average value of a sine wave is taken to mean the average of the full-wave rectified waveform. The RMS value of a sine wave is equal to *1.11 times* the sine wave average amplitude. *This relationship does not hold true for other waveshapes.*

dBm The decibel (*dB*) is commonly used to describe RMS voltage or power ratios between two signals. By itself, a decibel value has no particular meaning. Decibels are a ratio or comparison unit and have no absolute meaning without knowledge of a reference or comparison unit. When power comparisons are made to a 1 mW reference level, the letter *m* is added to give "dB*m*". For power ratios such as dBm, it is common to specify the resistance loading the voltage source. Often the system resistance is added to the units by indicating "dBm (50Ω)" for a 50Ω resistance system.

$$dB = 10 \times \log_{10} (P / Pref)$$

$$dBm = 10 \times \log_{10} (P / 0.001)$$

where power $P = V^2/R$

For a 50Ω resistance, 1 mW of power corresponds to 0.224 VRMS.

dBm (50Ω)	Output Voltage Level (50Ω load)
+23.98	3.53 VRMS
+13.01	1.00 VRMS
+6.99	500 mVRMS
0.0	224 mVRMS
-6.99	100 mVRMS
-13.01	50 mVRMS

Use the following conversions to determine dBm levels when connecting 75Ω or 600Ω load resistances.

$$dBm\ (75\ \Omega) = dBm\ (50\ \Omega) - 1.75$$

$$dBm\ (600\ \Omega) = dBm\ (50\ \Omega) - 10.79$$

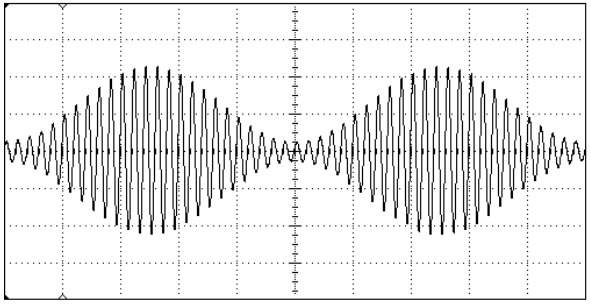
Modulation

Modulation is the process of combining a high-frequency carrier signal and a low-frequency information signal. How these signals are combined is determined by the specific type of modulation used. The two most common types of modulation are *amplitude modulation* (*AM*) and *frequency modulation* (*FM*). The information signal that modulates (or varies) the carrier waveform can be of any form — sine wave, square wave, arbitrary wave, or random noise. In general, the carrier signal may also be of any shape, but it is usually a sine wave of constant amplitude and frequency for most communications systems. During modulation, the simple carrier waveform is converted into a complex waveform by the lower-frequency information signal. Generally, the higher-frequency carrier waveform is used to efficiently transmit the complex modulated signal over long distances.

Amplitude Modulation (AM) Amplitude Modulation is a process of producing a waveform whose amplitude varies as a function of the instantaneous amplitude of the modulating information signal. In other words, the information signal creates an amplitude "envelope" around the carrier signal. The 33120A implements "double sideband transmitted carrier" amplitude modulation similar to a typical AM radio station. A constant is added to the AM modulating signal so that the sum is always greater than zero (for <100% modulation) as this equation shows:

$$\frac{(1 + D \times Am(t)) \times \sin(2\pi \times Fc \times t)}{2}$$

"D" is the modulation depth ($0 \le D \le 1.2$).
"Am" is the modulating signal with peak amplitude ≤ 1.
"Fc" is the carrier frequency.



An AM waveform with 80% modulation depth. The carrier waveform is a 5 kHz sine wave and the modulating waveform is a 200 Hz sine wave.

In amplitude modulation, the amplitude of the carrier varies between zero and twice its normal value for 100% modulation. The *percent modulation depth* is the ratio of the peak information signal amplitude to the constant. ***When amplitude modulation is selected, the 33120A automatically reduces its peak-to-peak amplitude by one-half so that a 100% modulation depth signal can be output***. Amplitude settings are defined to set the 100% peak-to-peak amplitude independent of the modulation depth setting. *Vrms and dBm amplitude settings are not accurate in AM since signals are very complex.*

Frequency Modulation (FM) Frequency Modulation is a process of producing a wave whose frequency varies as a function of the instantaneous amplitude of the modulating information signal. The extent of carrier frequency change is called *deviation*. The frequency deviations are caused by the amplitude changes of the modulating information signal. You can set the amount of the peak frequency in FM with the *deviation* parameter.

In frequency modulation, "100% modulation" has a different meaning than in AM. Modulation of 100% in FM indicates a variation of the carrier by the amount of the full permissible deviation. Since the modulating signal only varies frequency, the amplitude of the signal remains constant regardless of the modulation. The function generator uses the *deviation* parameter to describe the peak frequency change above or below the carrier in response to a corresponding amplitude peak of the modulating signal. For FM signals, the bandwidth of the modulated signal can be approximated by:

BW ≈ 2 x (Deviation + Information Signal Bandwidth) *For wideband FM*

BW ≈ 2 x (Information Signal Bandwidth) *For narrowband FM*

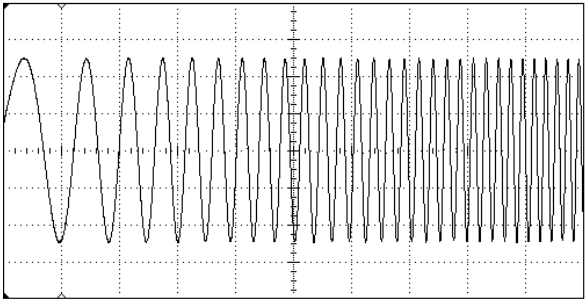
Narrowband FM occurs when the ratio of the deviation frequency to the information signal bandwidth is approximately 0.01 or less. Wideband commercial FM radio stations in the United States use a 75 kHz peak deviation (150 kHz peak-to-peak) and audio signals band-limited to 15 kHz to achieve 200 kHz channel-to-channel spacing from the 180 kHz bandwidth.

Frequency Sweep The 33120A performs phase-continuous frequency sweeping — stepping from the sweep start frequency to the sweep stop frequency with between 2,048 and 4,096 discrete frequency steps. The direction of frequency sweeps can be varied by setting the stop frequency either above or below the start frequency. Individual frequency steps are either linearly or logarithmically spaced based on the *sweep mode* setting. Like FSK modulation (described on the next page), the sweep function is also a special case of frequency modulation (FM). All of the FM operations described on the previous page also apply to sweep when the following translations are applied:

$$\text{Carrier Frequency} = \frac{\text{Start Frequency} + \text{Stop Frequency}}{2}$$

$$\text{Deviation} = \frac{\text{Start Frequency} - \text{Stop Frequency}}{2}$$

The modulation waveshape for sweeps is a ramp wave or exponential wave for linear or log sweeps, respectively. The logic sense of the ramp or exponential modulation signal (positive or negative ramp) is selected when the stop frequency is either larger or smaller than the start frequency. Like the FM function, changes to sweep parameters cause the generator to automatically compute a modulation signal and download it into modulation RAM. Similarly, the *sweep time* parameter adjusts the period of the modulating waveform. The sweep function also allows triggered operation. This is like frequency modulating with a single cycle burst of the modulating signal beginning when a trigger is received. Trigger signals can come from the rear-panel *Ext Trig* terminal, the front-panel Single button, or from commands over the remote interface.

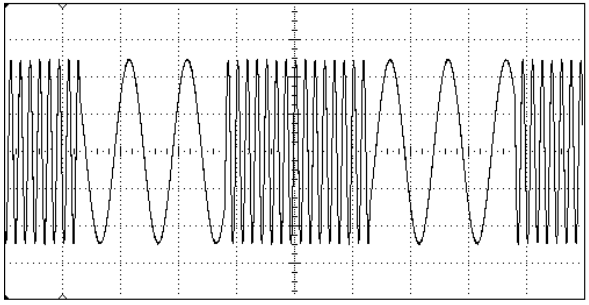


A sine wave sweep from 50 Hz to 5 kHz with a linear 1 second sweep time.

Frequency Shift Key Modulation In Frequency-Shift Keying modulation (*FSK*), the function generator's output frequency alternates between the carrier frequency and a second "hop" frequency. The rate of frequency hops is controlled either by an internal source or from an external logic input. FSK is essentially a special case of frequency modulation (FM) where the hop frequency is another way of specifying both the deviation and the modulating signal shape.

The modulating signal shape is always a square wave with an amplitude of zero to +1. The deviation is either positive or negative depending on whether the hop frequency is larger or smaller than the present carrier frequency (as shown below). The internal FSK rate generator specifies the period of the modulating square wave signal. When selected, the external FSK input replaces the internal FSK rate generator to directly control the frequency hop rate. A TTL "low" input always selects the carrier frequency and a TTL "high" always selects the hop frequency. The logic sense of the external FSK input can effectively be changed by interchanging the carrier and hop frequency values.

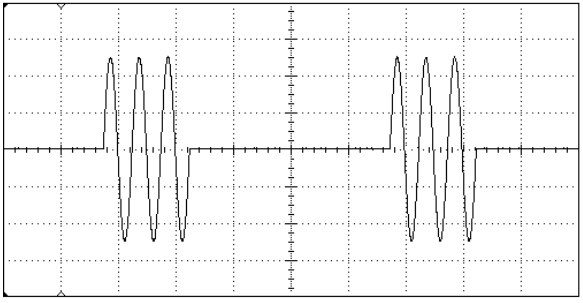
Deviation = Hop Frequency – Carrier Frequency



An FSK waveform with a 3 kHz carrier waveform and 500 Hz "hop" waveform (the FSK rate is 100 Hz).

Burst Modulation In burst modulation, the function generator turns the carrier wave output "on" and "off " in a controlled manner. The carrier output can be controlled using either triggered or externally-gated methods.

When configured for *triggered* operation, the function generator can output a carrier waveform with a user-specified number of complete cycles. Each time a trigger is received, the specified number of complete cycles is output. You can also specify a starting waveform phase in triggered operation. Zero degrees is defined as the first data point in waveform memory. The function generator will output the start phase as a dc output level while waiting for the next trigger. Output dc offset voltages are not affected by burst modulation — they are independently produced and summed into the function generator's output amplifier.



A three-cycle bursted sine wave with 100 Hz burst rate.

In *gated burst* mode operation, the rear-panel *Burst* terminal is used to directly (and asynchronously) turn off the waveform DAC output. The burst count, burst rate, and burst phase settings have no effect in this mode. When the burst signal is true (TTL "high"), the function generator outputs the carrier waveform continuously. When the burst signal is false (TTL "low"), the waveform DAC is forced to a zero output level. Like triggered burst operation, the output dc offset voltage is not affected by the external burst gate signal.

For *triggered burst* operation, the function generator creates an internal modulation signal which is exactly synchronous with the carrier waveform. This internal modulation signal is used to halt waveform memory addressing when the last data point is reached. This modulation signal effectively "gates" the output "on" and "off" for the specified number of carrier wave cycles. The modulation signal is then triggered by another internal burst rate signal generator which controls how often the specified carrier burst is output. In external triggered burst operation, the modulation signal trigger source is set to the function generator's rear-panel *Ext Trig* terminal. This source replaces the internal burst rate signal generator for pacing triggered bursts.

Changes to the burst count, burst rate, burst phase, or carrier frequency will cause the function generator to automatically compute a new modulation signal and download it into modulation RAM. *It is not possible for the function generator to burst single cycles for all carrier frequencies* because the internal modulation signal generator is not as capable as the main carrier signal generator. The table below shows the function generator's carrier frequency and burst count limitations.

Carrier Frequency	Minimum Burst Count
10 mHz to 1 MHz	1
>1 MHz to 2 MHz	2
>2 MHz to 3 MHz	3
>3 MHz to 4 MHz	4
>4 MHz to 5 MHz	5

For sine, square, and arbitrary waveforms only.

Internal Modulation Source Internally, the function generator incorporates a second, lower speed and lower resolution DDS arbitrary waveform generator to produce the modulating signal independent of the carrier signal. Internal modulation waveshapes range in length from 2,048 points to 4,096 points. User-defined arbitrary waveforms are automatically expanded or compressed in length as needed to fit within the required modulation waveform constraints. Linear interpolation is performed on user-defined arbitrary waveforms while the lengths of standard waveshapes are varied by decimation. Due to the modulation sample rate and waveform size limitations, the best case modulation signal frequency accuracy is approximately 0.05% of setting.

Unlike the main signal output discussed previously, modulation waveshapes are sampled using a variable “point clock” to sample data loaded in modulation waveform RAM. Internally, the modulation point clock (C) and modulation waveform length are automatically adjusted to produce the modulation signal frequency desired. For frequencies greater than C/2048, all modulation shapes are sampled up to the maximum modulating frequency. A new modulation waveform is computed and loaded into modulation RAM each time the modulation type, modulation waveshape, or modulation frequency is changed. Data in standard arbitrary waveform memory is not affected by modulation signal changes (data is expanded or compressed and loaded directly into separate modulation RAM following computation). No expansion or compression is performed on the modulation waveform data for certain modulation frequencies.

You can use the equations on the next page to determine specific waveform lengths and modulation frequencies when more precise control is needed. *Normally, you should not have to perform these calculations.*

The function generator incorporates an internal 8-bit (± 7 bits peak) digital-to-analog converter (DAC) to create an analog copy of the modulation signal for amplitude modulation (AM). This signal is internally applied to a conventional four-quadrant analog multiplier circuit to achieve amplitude modulation. Similarly, the generator uses digital signal processing to combine the carrier and modulation signals for frequency modulation (FM). The FM modulation signal maintains 12-bit resolution for frequency values.

The following equations and example describe the capabilities and limitations of the 33120A's internal modulation signal generator.

Parameter Definitions:

Maximum Point Clock (C) = 5 MSa/ s *(for AM)*
1.25 MSa/ s *(for FM)*

Modulation Prescaler (S) = integer numbers (truncated) from 1, 2, 3, ... 2^{20}

Constant (k) = 4,900 *(for AM)*
624 *(for FM)*

Modulation Frequency (F) = 10 mHz to 20 kHz *(for AM)*
10 mHz to 10 kHz *(for FM)*

Points (P) = values from 2,000 to 4,000
even numbers only (*rounded down*)

Equations:

Compute the modulation pre-scaler divider:

$$S = \frac{k}{F}$$ *(truncated to integer value ≥ 1)*

Compute the number of points for the modulation waveform length:

$$P = \frac{2 \times C}{(1 + S) \times F}$$ *(rounded down to even number)*

Waveshapes are automatically expanded or compressed to match length "P" computed above and downloaded into modulation RAM.

Example: Assume that you need to phase-continuously frequency hop between the following nine frequencies every 200 μs.

15.0 MHz, 1.001 MHz, 9.780 MHz, 12.375 MHz, 0.5695 MHz, 3.579 MHz, 0.8802 MHz, 0.6441 MHz, and 10.230 MHz.

Solution: Create a modulation arbitrary waveform that is precisely sampled in FM modulation.

$F = 1 / (9 \times 200\ \mu s) = 555.555$ Hz *(modulation frequency)*

Round down in sixth digit to get modulation frequency to set.

$S = 624 / 555.555 = 1.1232$ *(truncate to **1**)*

$$P = \frac{2 \times C}{(1 + S) \times F}$$ *(rounded down to even number)*

- Set the *modulation frequency* to 555.555 Hz.
- Set the *carrier frequency* to (Max F + Min F) / 2 = 7.784750 MHz.
- Set *deviation (pk) frequency* to (Max F – Min F) / 2 = 7.215250 MHz.
- Create and download a nine-segment arbitrary waveform with the values shown below. Each segment is 250 points long (2250/9) for a total of 2,250 points. Use the `DATA VOLATILE` command download to achieve 12-bit frequency resolution for each point.

$$y = mX + b$$

"y" is the new vertical value.
"m" = 1 / Deviation
"X" is the original frequency point.
"b" = – carrier frequency / Deviation = 1.078930044

Segment	Value
1	+1.0000
2	-0.9402
3	+0.2765
4	+0.6362
5	-1.0000
6	-0.5829
7	-0.9569
8	-0.9897
9	+0.3389

To Check: Enable FM by sending the following commands:

```
"FM:STATE ON"
"FM:INT:FREQ 555.555"
"DIAG:PEEK? 0,0,0"
enter results    < Prescale# (S) > , < Points (P) >
```

Specifications

WAVEFORMS

Standard Waveforms: Sine, Square, Triangle, Ramp, Noise, DC volts, Sine(x)/x, Negative Ramp, Exponential Rise, Exponential Fall, Cardiac

Arbitrary Waveforms:

Waveform Length:	8 to 16,000 points
Amplitude Resolution:	12 bits (including sign)
Sample Rate:	40 MSa / sec
Non-Volatile Memory:	Four 16,000-point waveforms

FREQUENCY CHARACTERISTICS

Sine:	100 μHz – 15 MHz
Square:	100 μHz – 15 MHz
Triangle:	100 μHz – 100 kHz
Ramp:	100 μHz – 100 kHz
Noise (Gaussian)**:**	10 MHz bandwidth
Arbitrary Waveforms:	
8 to 8,192 points:	100 μHz – 5 MHz
8,193 to 12,287 points:	100 μHz – 2.5 MHz
12,288 to 16,000 points:	100 μHz – 200 kHz

Resolution: 10 μHz or 10 digits

Accuracy: 10 ppm in 90 days, 20 ppm in 1 year, 18°C – 28°C

Temperature Coefficient: < 2 ppm / °C

Aging: < 10 ppm / yr

SINEWAVE SPECTRAL PURITY (into 50Ω)

Harmonic Distortion

DC to 20 kHz:	-70 dBc
20 kHz to 100 kHz:	-60 dBc
100 kHz to 1 MHz:	-45 dBc
1 MHz to 15 MHz:	-35 dBc

Total Harmonic Distortion

DC to 20 kHz:	< 0.04%

Spurious (non-harmonic)

Output (DC to 1 MHz):	< -65 dBc
Output (> 1 MHz):	< -65 dBc + 6 dB/octave

Phase Noise: < -55 dBc in a 30 kHz band

SIGNAL CHARACTERISTICS

Square wave

Rise/Fall Time:	< 20 ns
Overshoot:	< 4%
Asymmetry:	1% + 5 ns
Duty Cycle:	20% to 80% *(to 5 MHz)*
	40% to 60% *(to 15 MHz)*

Triangle, Ramp, Arb

Rise/Fall Time:	40 ns *(typical)*
Linearity:	< 0.1% of peak output
Settling Time:	< 250 ns to 0.5% of final value
Jitter:	< 25 ns

OUTPUT CHARACTERISTICS (1)

Amplitude (into 50Ω): (2)	50 mVpp – 10 Vpp
Accuracy (at 1 kHz):	± 1% of specified output
Flatness	(sine wave relative to 1 kHz)
< 100 kHz:	± 1% (0.1 dB)
100 kHz to 1 MHz:	± 1.5% (0.15 dB)
1 MHz to 15 MHz:	± 2% (0.2 dB) Ampl ≥ 3Vrms
1 MHz to 15 MHz:	± 3.5% (0.3 dB) Ampl < 3Vrms
Offset (into 50Ω): (3)	± 5 Vpk ac + dc
Accuracy: (4)	± 2% of setting + 2 mV

Output Impedance: 50 ohms fixed

Resolution: 3 digits, Amplitude and Offset

Output Units: Vpp, Vrms, dBm

Isolation: 42 Vpk maximum to earth

Protection: Short-circuit protected
± 15 Vpk overdrive < 1 minute

(1) Add 1/10th of output amplitude and offset specification per °C for operation outside of 18°C to 28°C range (1-year specification).

(2) 100 mVpp – 20 Vpp amplitude into open-circuit load.

(3) Offset ≤ 2 X peak-to-peak amplitude.

(4) For square wave outputs, add 2% of output amplitude additional error.

MODULATION CHARACTERISTICS

AM Modulation

Carrier -3 dB Freq:	10 MHz *(typical)*
Modulation:	Any internal waveform plus Arb
Frequency:	10 mHz to 20 kHz (± 0.05% to 2.5 kHz, then decreases linearly to ± 0.4% at upper limit)
Depth:	0% to 120%
Source:	Internal / External

FM Modulation

Modulation:	Any internal waveform plus Arb
Frequency:	10 mHz to 10 kHz (± 0.05% to 600 Hz, then decreases linearly to ± 0.8% at upper limit)
Peak Deviation:	10 mHz to 15 MHz
Source:	Internal Only

Burst Modulation

Carrier Frequency:	5 MHz max.
Count:	1 to 50,000 cycles, or Infinite
Start Phase:	-360° to +360°
Internal Rate:	10 mHz to 50 kHz ±1%
Gate Source:	Internal or External Gate (1)
Trigger Source:	Single, External, or Internal Rate

FSK Modulation

Frequency Range:	10 mHz to 15 MHz (± 0.05% to 600 Hz, then decreases linearly to ± 4% at upper limit)
Internal Rate:	10 mHz to 50 kHz
Source:	Internal / External (1 MHz max.)

FREQUENCY SWEEP

Type:	Linear or Logarithmic
Direction:	Up or Down
Start F / Stop F:	10 mHz to 15 MHz
Time:	1 ms to 500 sec ±0.1%
Source:	Single, External, or Internal

REAR-PANEL INPUTS

External AM Modulation:	± 5 Vpk = 100% Modulation 5 kΩ Input Resistance
External Trigger/FSK Burst Gate: (1)	TTL (low true)
Latency:	1.3 μs
Jitter:	25 ns

SYSTEM CHARACTERISTICS

Configuration Times (2)

Function Change: (3)	80 ms
Frequency Change: (3)	30 ms
Amplitude Change:	30 ms
Offset Change:	10 ms
Select User Arb:	100 ms
Modulation Parameter Change:	< 350 ms

Arb Download Times over GPIB:

Arb Length	*Binary*	*ASCII Integer*	*ASCII Real* (4)
16,000 points	8 sec	81 sec	100 sec
8,192 points	4 sec	42 sec	51 sec
4,096 points	2.5 sec	21 sec	26 sec
2,048 points	1.5 sec	11 sec	13 sec

Arb Download Times over RS-232 at 9600 Baud: (5)

Arb Length	*Binary*	*ASCII Integer*	*ASCII Real* (6)
16,000 points	35 sec	101 sec	134 sec
8,192 points	18 sec	52 sec	69 sec
4,096 points	10 sec	27 sec	35 sec
2,048 points	6 sec	14 sec	18 sec

(1) Trigger source ignored when External Gate is selected.

(2) Time to change parameter and output the new signal.

(3) Modulation or sweep off.

(4) Times for 5-digit and 12-digit numbers.

(5) For 4800 baud, multiply the download times by two; For 2400 baud, multiply the download times by four, etc.

(6) Time for 5-digit numbers. For 12-digit numbers, multiply the 5-digit numbers by two.

GENERAL SPECIFICATIONS

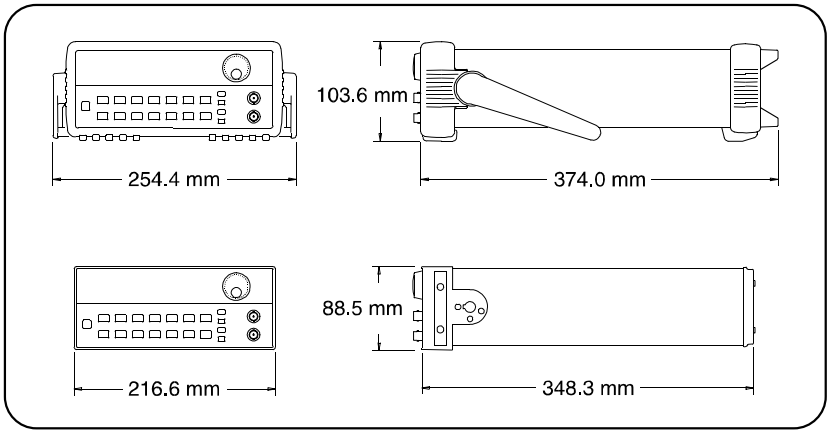
Power Supply: (1)	100V / 120V / 220V / 240V ±10% *(switch selectable)*
Power-Line Frequency:	50 Hz to 60 Hz ±10% and 400 Hz ±10%. Automatically sensed at power-on.
Power Installation:	CAT II
Power Consumption:	50 VA peak (28 W average)
Operating Environment:	0°C to 55°C 80% Relative Humidity to 40°C Indoor or sheltered location
Storage Environment:	-40°C to 70°C
State Storage Memory:	Power-off state automatically saved. Three (3) User-Configurable Stored States, Arbitrary waveforms stored separately.
Dimensions (W x H x D)	
Bench Top:	254.4 mm x 103.6 mm x 374 mm
Rack Mount:	212.6 mm x 88.5 mm x 348.3 mm
Weight:	4 kg (8.8 lbs)

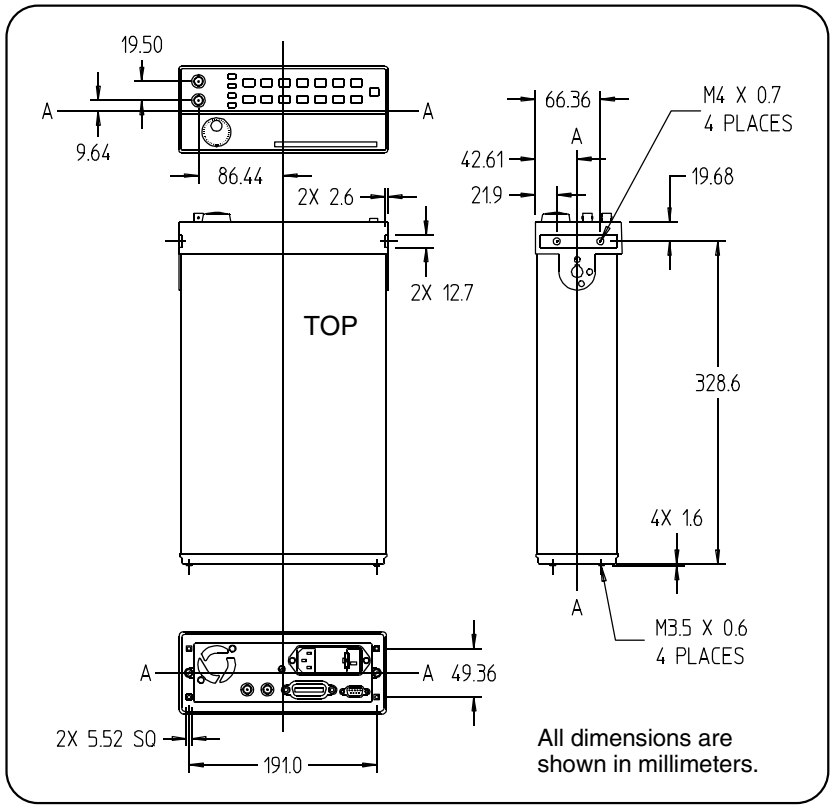
(1) For 400 Hz operation at 120 Vac, use the 100 Vac line-voltage setting.

Safety Designed to:	EN61010, CSA1010, UL-1244
EMC:	EN61326, 1:1997 + 1A:1998
Vibration and Shock:	MIL-T-28800E, Type III, Class 5 (data on file)
Acoustic Noise:	30 dBa
Warm-Up Time:	1 hour
Warranty:	3 years standard
Remote Interface:	IEEE-488 and RS-232 standard
Programming Language:	SCPI-1993, IEEE-488.2
Accessories Included:	User's Guide, Service Guide, Quick Reference Card, IntuiLink Arb software, RS-232 cable, Test Report, and power cord.

N10149

PRODUCT DIMENSIONS





Index

*If you have questions relating to the operation of the function generator, call **1-800-452-4844** in the United States, or contact your nearest Agilent Sales Office.*

*CLS command, 209
*ESE/*ESE? command, 209
*ESR? command, 209
*IDN? command, 113, 190
*OPC/*OPC? command, 166, 173, 210
*PSC/*PSC? command, 210
*RCL command, 153
*RST command, 171, 190
*SAV command, 152
*SRE/*SRE? command, 210
*STB? command, 210
*TRG command, 173, 187
*TST? command, 190
*WAI command, 166, 173, 210
34398A Cable Kit, 197
34399A Adapter Kit, 196

A
ac line voltage
  fuse, 16
  selecting, 16
ac signals attributes, 283
adapters (RS-232), 196
address (GPIB), 114, 217
Adrs annunciator, 5
AM annunciator, 5
AM modulation
  carrier frequency, 72
  carrier waveshape, 72
  example program, 252
  external modulation, 75, 156
  front-panel operation, 41
  modulating frequency, 74, 155
  modulating source, 75, 156
  modulating waveshape, 73, 155
  modulation depth, 74, 155
  operational overview, 154
  rear-panel terminal, 75, 155
  sync signal, 74
  technical description, 287
AM:DEPT command, 155
AM:INT:FREQ command, 155
AM:INT:FUNC command, 155
AM:SOUR command, 156
AM:STAT command, 156
amplitude
  and output termination, 60, 149
  arb waveform limits, 148, 177
  conflict with function, 59, 149
  front-panel selection, 20
  limits, 59, 148
  modulation, 71, 154
  remote interface selection, 148
  restrictions with offset, 60, 149
  selecting units, 64, 150
angle, bursts, 89, 163
annunciators, front panel, 5
application programs, 243 - 269
APPLy command, 136, 138 - 144
  MIN/MAX parameters, 143
  optional parameters, 143
Arb annunciator, 5
arbitrary waveforms
  amplitude limitations, 148, 177
  average value, 181
  binary data, 179
  built-in waveforms, 175
  catalog listing, 182, 184
  copy name, 182
  crest factor value, 181
  delete waveform, 183
  download speeds, 299
  example program (GPIB), 255
  example program (RS-232), 267
  floating-point data, 178
  frequency limits, 57, 146
  front-panel operation, 103 - 108
  inherent offset, 63, 177
  memory available, 185
  memory limitations, 183
  modulation matrix, 56, 177
  naming rules, 176, 182
  number of points, 181
  parity and data bits, 116, 195
  peak-to-peak value, 181
  remote operation, 174 - 185
  reserved names, 182
  tutorial overview, 278
average value of waveform, 181, 285

B
BASIC examples, 244
baud rate (RS-232)
  choices, 116, 195
  factory setting, 116, 195
  selecting, 219
beep, 189
binary data
  arb waveforms, 179
  block format, 180
  byte order, 185
bit definitions
  standard event register, 207
  status byte, 203
block format, binary data, 180
blue Shift key, 14
BM:INT:RATE command, 164
BM:NCYC command, 162
BM:PHAS command, 163
BM:SOUR command, 164
BM:STAT command, 165
Borland® Turbo C®++, 244
braces ({ }), syntax, 127
brackets ([ ]), syntax, 127
built-in arb waveforms, 175
bumpers, removing, 27
Burst annunciator, 5
burst modulation
  burst count restrictions, 85, 162
  burst count selection, 87, 162
  carrier frequency range, 84, 162
  example program, 248
  external gate source, 83, 164
  front-panel operation, 47
  gated mode, 81, 160
  internal rate, 88, 164
  starting phase, 89, 163
  sync signal, 86
  technical description, 291
  trigger source, 82, 165
  triggered mode, 81, 160
Burst terminal, 102
bus triggering, 101, 186
byte order (binary data), 180, 185

C

C examples, 244
cables (RS-232), 196
CAL:COUN? command, 193
CAL:SEC:CODE command, 193
CAL:SEC:STAT command, 193
CAL:SET command, 194
CAL:STR command, 194
CAL:VAL command, 194
CAL? command, 193
calibration
  overview, 118 - 122
  read count, 121, 193
  security code, 118
  text message, 122
  to secure, 120
  to unsecure, 119
capability codes (GPIB), 225
cardiac waveform, 175
carriage return character, 214
carrier frequency
  AM, 72
  burst modulation, 84
  FM, 77
  FSK, 91
carrying handle
  adjustment, 18
  changing positions, 18
  removing, 27
catalog of arb names, 182, 184
chassis ground, 6
colon (:), syntax, 213
COM (serial) ports, 199
command format (SCPI), 212
command syntax
  colon vs semicolon, 213
  conventions, 127
  terminators (SCPI), 214
commas in display, 52, 113
common commands (IEEE), 214, 225
computer connection (RS-232), 196
contents, 9 - 11
crest factor (arb data), 181, 284
Ctrl-C character, 199
cycles (burst), 87, 162

D

damped sine wave, example, 267
data bits (RS-232)
  arb waveform limits, 116, 195
  choices, 116, 195
  factory setting, 116, 195
  selecting, 220
data frame (RS-232), 196
data points, arb waveforms, 181
DATA VOLATILE command, 178
DATA:ATTR:AVER? command, 181
DATA:ATTR:CFAC? command, 181
DATA:ATTR:POIN? command, 181
DATA:ATTR:PTP? command, 181
DATA:CAT? command, 182
DATA:COPY command, 182
DATA:DAC VOLATILE command, 179
DATA:DEL command, 183
DATA:DEL:ALL command, 184
DATA:NVOL:CAT? command, 184
DATA:NVOL:FREE? command, 185
dBm, 64, 150, 286
dc offset
  and arb waveforms, 63, 177
  and dc voltage function, 62, 150
  and output termination, 62, 150
  front-panel selection, 21
  remote interface selection, 149
dc volts
  front-panel selection, 24
  selecting, 62, 150
  with APPLy, 144
declaration of conformity, 309
deviation, peak frequency, 80, 158
device clear, 216
digits, maximum number, 112
dimensions, product, 301
direct digital synthesis, 273
DISP command, 188
DISP:TEXT command, 188
DISP:TEXT:CLE command, 189
display
  annunciators, 5
  enable/disable, 112, 188
  text message, 112, 188
download speeds
  arb waveforms (GPIB), 299
  arb waveforms (RS-232), 299
downloading arb data
  binary data, 179
  floating-point data, 178
DSR line (RS-232), 197, 198
DTR line (RS-232), 197, 198
duty cycle
  conflict with frequency, 67, 147
  definition, 66
  front-panel selection, 22
  remote interface selection, 147

E

"Enter Number" mode, 3, 38
ERROR annunciator, 5
error interrupt, example, 261
errors, 227 - 241
  clearing, 110, 228
  reading error queue, 110, 228
even parity (RS-232), 116, 220
example programs (remote interface)
  AM waveform, 252
  APPLy command, 248
  arb waveform (GPIB), 255
  arb waveform (RS-232), 267
  burst modulation, 248
  damped sine wave, 267
  error interrupt, 261
  haversine, 248
  state storage, 252
  status registers, 261
  triggering, 248
exponential fall waveform, 175
exponential rise waveform, 175
Ext annunciator, 5
Ext Trig terminal, 81, 102, 160, 187
external source
  AM, 75, 156
  burst modulation, 82, 164
  FSK, 93, 169
  sweep, 97, 173
external triggering, 100, 186

F
factory setting
baud rate (RS-232), 116, 219
calibration security code, 118
data bits (RS-232), 116, 220
GPIB address, 114, 217
parity (RS-232), 116, 220
firmware revision, 113, 190
floating-point data, 178
FM annunciator, 5
FM modulation
carrier frequency, 78
carrier waveshape, 77
modulating frequency, 79, 159
modulating waveshape, 79, 159
operational overview, 157
peak frequency deviation, 80, 158
sync signal, 79
technical description, 288
FM:DEV command, 158
FM:INT:FREQ command, 159
FM:INT:FUNC command, 159
FM:STAT command, 159
FORM:BORD command, 185
FREQ command, 146
FREQ:STAR command, 171
FREQ:STOP command, 171
frequency deviation (FM), 80, 158
frequency sweep
See sweep
frequency
arb waveform limits, 57, 146
conflict with duty cycle, 58, 147
conflict with function, 58, 146
front-panel selection, 19
limits, 57, 146
modulation, 76, 157
remote interface selection, 57, 146
frequency-shift keying
See FSK
front panel
annunciators, 5
dc volts selection, 24
downloading waveforms, 103 - 108
menu examples, 33 - 39
menu messages, 34
menu reference, 31 - 32
number entry, 3, 38
overview, 2
front-panel display
annunciators, 5
enable/disable, 112, 188
text message, 112, 188
FSK annunciator, 5
FSK terminal, 102
FSK
carrier frequency, 91
front-panel operation, 44
"hop" frequency, 92, 168
internal rate, 93, 168
maximum external rate, 93
operational overview, 167
source, 93, 169
sync signal, 91
technical description, 290
FSK:FREQ command, 168
FSK:INT:RATE command, 168
FSK:SOUR command, 169
FSK:STAT command, 169
FUNC:SHAP command, 145
FUNC:SHAP USER command, 177
FUNC:USER command, 176
function
conflict with amplitude, 55
conflict with frequency, 55
front-panel selection, 56
modulation matrix, 56, 145
remote interface selection, 145
fuse, power line, 16

G
gate source
burst modulation, 164
gated burst mode, 81, 160
GPIB
arb waveform download, 255
capability codes, 225
download speeds, 299
selecting interface, 115, 218
GPIB address
factory setting, 114, 217
selecting, 114, 217
GPIB Command Library, 244
ground loops, 282
ground, chassis, 6
Group Execute Trigger, 101, 186

H
handle
adjustment, 18
changing positions, 18
removing, 27
handshake (RS-232), 198
haversine, example, 248
heartbeat waveform, 175
hop frequency (FSK), 92, 168
HP-IB address
See GPIB address

I
identification string, 113, 190
IEEE-488
common commands, 214, 225
conformance information, 225
selecting interface, 115, 218
IEEE-488 address
factory setting, 114, 217
selecting, 114, 217
inherent offset, 63, 177
interface address
factory setting, 114, 217
selecting, 114, 217
IntuiLink software, 1

interface (remote)
selecting, 115, 218
triggering, 101, 186
internal burst rate, 88, 164
internal FSK rate, 93, 168
internal triggering, 99, 186
inventory checklist, 15
IOTRIGGER command, 101

L
language
interface query, 117, 190
SCPI introduction, 211 - 215
line voltage
fuse, 16
selecting, 16
linear spacing (sweep), 96, 172
load (output termination)
front-panel selection, 40, 65
remote interface selection, 65, 151
schematic diagram, 281
logarithmic spacing (sweep), 96, 172
logic levels
Ext Trig/FSK/Burst terminal, 102
long form, SCPI commands, 212
low-level commands, 136

M
manual overview, 7
MEM:STAT:DEL command, 153
memory available (arbs), 185
menu (front-panel)
examples, 33 - 39
menu recall, 33, 37
messages, 34
operation, 29
overview, 4
reference, 31 - 32
message
calibration, 122
errors, 227 - 241
front-panel display, 112, 188
message available, 205
Microsoft® Quick C®, 244
MIN/MAX parameters, 143, 213
modulating frequency
AM, 74, 155
FM, 79, 159
FSK, 92, 168
modulating source (AM), 75, 156
modulating waveshape
AM, 73, 155
FM, 79, 159
modulation depth (AM), 74, 155
with external source, 75, 155

N
negative ramp waveform, 175
new line character, 214
Num annunciator, 5
"Number Entry" mode, 3, 38
number of cycles (burst), 87, 162
Nyquist Sampling Theorem, 276

O
odd parity, 116, 220
Offset annunciator, 5
offset
and arb waveforms, 63, 177
and dc voltage function, 62, 150
and output termination, 62, 150
front-panel selection, 21
remote interface selection, 149
Option 001 (Phase-Lock)
See Option 001 User's/Service Guide
optional parameters (APPLy), 143
OUTP:LOAD command, 151
OUTP:SYNC command, 151
output amplitude
and output termination, 60, 149
arb waveform limits, 148, 177
conflict with function, 59, 149
front-panel selection, 20
limits, 59, 148
remote interface selection, 148
restrictions with offset, 60, 149
selecting units, 64, 150
output configuration, 145 - 153
output frequency
arb waveform limits, 57, 146
conflict with duty cycle, 58, 147
conflict with function, 58, 146
front-panel selection, 19
limits, 57, 146
modulation, 76, 157
remote interface selection, 57, 146
output function
conflict with amplitude, 55
conflict with frequency, 55
front-panel selection, 56
modulation matrix, 56, 145
remote interface selection, 145
output termination (load)
front-panel selection, 40, 65
remote interface selection, 65, 151
schematic diagram, 281
output units
front-panel selection, 64
remote interface selection, 150
overview
front panel, 2
menu structure, 4
product, 1
rear panel, 6

P
parameter types (SCPI), 215
parity (RS-232)
arb waveform limits, 116, 195
choices, 116, 195
factory setting, 116, 195
selecting, 220
peak deviation (FM), 80, 158
peak-to-peak value, 181, 285
peak value, 285
percent modulation
See modulation depth
percentage
AM depth, 155
duty cycle, 147
phase (burst), 89, 163
phase-lock (Option 001)
See Option 001 User's/Service Guide

points, number in arb waveform, 181
ports, serial (COM), 199
power-down recall, 109, 152
power-line voltage
  fuse, 16
  selecting, 16
power-on state, 123
printing history, *inside front cover*
product dimensions, 301
product overview, 1
programming
  examples, 243 - 269
  overview, 136
  SCPI introduction, 211 - 215
PULS:DCYC command, 147
pulse, example program, 255

Q
query commands, 214
query, reading, 137
quick start, 13
QuickBASIC examples, 247
QuickC examples, 244

R
rack mounting kits, 27
rear panel
  Ext Trig/FSK/Burst terminal, 102
  overview, 6
recall menu, 33, 37
recall state
  front-panel operation, 26
  remote interface operation, 153
register diagram (status), 202
remote interface
  command reference, 125 - 225
  command summary, 127 - 135
  programming overview, 136
  selecting, 115, 218
  triggering, 101, 186
reserved names (arb waveforms), 182
reset operation, 190
reset state, 123
revision number (firmware), 113, 190
RMS value, 285
Rmt annunciator, 5
RS-232
  arb download example, 267
  baud rate, 116, 219
  cables and adapters, 196
  configuration, 195
  connection to computer, 196
  data frame format, 196
  download speeds for arbs, 299
  handshake, 198
  parity and data bits, 116, 220
  selecting interface, 115, 218
  start bits, 195
  stop bits, 195
  troubleshooting, 199
rubber bumpers, removing, 27
rules, arb waveform names, 176, 182

S
safety information, *inside front cover*
SCPI language
  command format, 212
  command summary, 127 - 135
  conformance information, 221
  introduction, 211 - 215
  language query, 117, 190
  long form syntax, 212
  parameter types, 215
  programming overview, 136
  short form syntax, 212
  status registers, 201 - 208
  syntax conventions, 127
  terminators, 214
  version query, 117, 190
security code (calibration)
  factory setting, 118
  to change, 121
self test
  complete, 111, 190
  power-on, 111
semicolon (;), syntax, 213
serial (COM) ports, 199
serial poll, 204
service request (SRQ), 204
Shift annunciator, 5
Shift key, 14
shipping inventory, 15
short form, SCPI commands, 212
sinc waveform, 175
single triggering, 100
sliding rack kit, 28
software, IntuiLink, 1
software triggering, 101, 186
spacing
  linear sweep, 96, 172
  logarithmic sweep, 96, 172
specifications, 297 - 301
speeds
  arb waveforms (GPIB), 299
  arb waveforms (RS-232), 299
standard event register
  bit definitions, 207
  clearing bits, 208
start bits (RS-232), 195
start frequency (sweeps), 95, 171
starting phase (bursts), 89, 163
state storage
  delete state, 153, 192
  example program, 252
  front-panel operation, 25
  power-down recall, 109
  recalling stored state, 153, 192
  remote interface operation, 152
state, power-on and reset, 123
status byte
  bit definitions, 203
  clearing bits, 203
status registers, 201 - 208
  example program, 261
  register diagram, 202
  standard event, 207
  status byte, 203
stop bits (RS-232), 195
stop frequency (sweeps), 95, 171
stored states
  delete state, 153, 192
  example program, 252
  front-panel operation, 25
  power-down recall, 109
  recalling stored state, 153, 192
  remote interface operation, 152

string
calibration, 122
display message, 112
swapped byte order, 185
SWE:SPAC command, 172
SWE:TIME command, 172
sweep
front-panel operation, 49, 94
linear spacing, 96, 172
logarithmic spacing, 96, 172
remote interface operation, 170
start frequency, 95, 171
stop frequency, 95, 171
sweep time, 96, 172
sync signal, 95
technical description, 289
trigger source, 97, 173
Swp annunciator, 5
sync signal
AM, 74
burst (gated mode), 86
burst (triggered mode), 86
enabling/disabling, 68, 151
FM, 79
for all functions, 68
FSK, 91
sweep, 95
SYNC terminal, 68, 151
syntax
colon vs semicolon, 213
SCPI conventions, 127
SYST:BEEP command, 189
SYST:ERR? command, 189, 209
SYST:LOC command, 200
SYST:REM command, 200
SYST:RWL command, 200
SYST:VERS? command, 190

T
table of contents, 9 - 11
technical specifications, 297 - 301
terminals
Ext Trig/FSK/Burst, 102, 187
front-panel overview, 2
rear panel overview, 6
termination (output load)
front-panel selection, 40, 65
remote interface selection, 65, 151
schematic diagram, 281
text message
calibration, 122
errors, 227 - 241
front-panel display, 112, 188
times for download
arb waveforms (GPIB), 299
arb waveforms (RS-232), 299
tone, 189
triangle brackets (< >), syntax 127
Trig annunciator, 5
TRIG:SOUR command, 165, 173, 186
TRIG:SLOP command, 187
TRIGGER command, 101, 186
triggering
burst modulation, 81, 165
example program, 248
external source, 100, 186
from the front panel, 51
from the remote interface, 186
internal, 99, 186
software (bus), 101, 186
troubleshooting
error messages, 227 - 241
power-on failure, 16
RS-232, 199
Turbo C, 244

U
units (output amplitude)
front-panel selection, 64
remote interface selection, 150

V
vertical bar (|), syntax, 127
VOLT command, 148
VOLT:OFFS command, 149
VOLT:UNIT command, 150
Vpk, 64, 150
Vpp, 64, 150
Vrms, 64, 150

W
warranty information, *inside front cover*
waveforms
ac attributes, 283
arbitrary download, 103, 174
built-in arbs, 175
download speeds (arbs), 299
selecting amplitude, 59, 148
selecting duty cycle, 66, 147
selecting frequency, 57, 146
selecting function, 55, 145
selecting load, 65, 151
selecting offset, 62, 149
selecting output units, 64, 150
specifications, 297 - 301
sync signal, 68
weight, product, 300
wiring adapter (RS-232), 196

Agilent Technologies	**DECLARATION OF CONFORMITY** According to ISO/IEC Guide 22 and CEN/CENELEC EN 45014	CE

Manufacturer's Name:	Agilent Technologies, Incorporated	Agilent Technologies (Malaysia) Sdn. Bhd.
Manufacturer's Address:	815 - 14th St. SW Loveland, Colorado 80537 USA	Bayan Lepas Free Industrial Zone 11900 Penang Malaysia

Declares that the product

Product Name:	15 MHz Function/Arbitrary Waveform Generator
Model Number:	33120A
Product Options:	This declaration covers all options of the above product.

***Conforms with the following European Directives*:**

The product herewith complies with the requirements of the Low Voltage Directive 73/23/EEC and the EMC Directive 89/336/EEC (including 93/68/EEC) and carries the CE Marking accordingly.

Conforms with the following product standards:

EMC	**Standard**	**Limit**
	IEC 61326-1:1997+A1:1998 / EN 61326-1:1997+A1:1998	
	CISPR 11:1990 / EN 55011:1991	*Group 1 Class A*
	IEC 61000-4-2:1995+A1:1998 / EN 61000-4-2:1995	*4kV CD, 8kV AD*
	IEC 61000-4-3:1995 / EN 61000-4-3:1995	*3 V/m, 80-1000 MHz*
	IEC 61000-4-4:1995 / EN 61000-4-4:1995	*0.5kV signal lines, 1kV power lines*
	IEC 61000-4-5:1995 / EN 61000-4-5:1995	*0.5 kV line-line, 1 kV line-ground*
	IEC 61000-4-6:1996 / EN 61000-4-6:1996	*3V, 0.15-80 MHz I cycle, 100%*
	IEC 61000-4-11:1994 / EN 61000-4-11:1994	*Dips: 30% 10ms; 60% 100ms* *Interrupt > 95%@5000ms*
	Canada: ICES-001:1998 *Australia/New Zealand: AS/NZS 2064.1*	

The product was tested in a typical configuration with Agilent Technologies test systems.

Safety *IEC 61010-1:1990+A1:1992+A2:1995 / EN 61010-1:1993+A2:1995*
Canada: CSA C22.2 No. 1010.1:1992
UL 3111-1: 1994

1 May 2001
Date

Ray Corson
Product Regulations Program Manager

For further information, please contact your local Agilent Technologies sales office, agent or distributor.
Authorized EU-representative: Agilent Technologies Deutschland GmbH, Herrenberger Strabe 130, D71034 Böblingen, Germany

© Copyright Agilent Technologies, Inc. 1994-2002

No part of this manual may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Agilent Technologies as governed by the United States and international copyright laws.

Manual Part Number

33120-90006, March 2002
(*order as 33120-90104 manual set*)

Edition

Edition 6, March 2002
Edition 5, August 1997
Edition 4, February 1996
Edition 3, May 1994
Edition 2, March 1994
Edition 1, January 1994

Printed in Malaysia

Agilent Technologies, Inc.
815 14th Street S.W.
Loveland, Colorado 80537 U.S.A.

Assistance

Product maintenance agreements and other customer assistance agreements are available for Agilent Technologies products. For assistance, contact your nearest Agilent Technologies Sales and Service Office. Further information is available on the Agilent web site at *www.agilent.com/find/assist*.

Trademark Information

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation. All other brand and product names are trademarks or registered trademarks of their respective companies.

Certification

Agilent Technologies certifies that this product met its published specifications at the time of shipment from the factory. Agilent Technologies further certifies that its calibration measurements are traceable to the United States National Institute of Standards and Technology, to the extent allowed by that organization's calibration facility, and to the calibration facilities of other International Standards Organization members.

Warranty

The material contained in this document is provided "as is," and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Agilent disclaims all warranties, either express or implied with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Agilent shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or any information contained herein. Should Agilent and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement will control.

Technologies Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

Restricted Rights Legend

If software is for use in the performance of a U.S. Government prime contract or subcontract, Software is delivered and licensed as "Commercial computer software" as defined in DFAR 252.227-7014 (June 1995), or as a "commercial item" as defined in FAR 2.101(a) or as "Restricted computer software" as defined in FAR 52.227-19 (June 1987) or any equivalent agency regulation or contract clause. Use, duplication or disclosure of Software is subject to Agilent Technologies' standard commercial license terms, and non-DOD Departments and Agencies of the U.S. Government will receive no greater than Restricted Rights as defined in FAR 52.227-19(c)(1-2) (June 1987). U.S. Government users will receive no greater than Limited Rights as defined in FAR 52.227-14 (June 1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

Safety Notices

Do not install substitute parts or perform any unauthorized modification to the product. Return the product to an Agilent Technologies Sales and Service Office for service and repair to ensure that safety features are maintained.

WARNING

A WARNING notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in personal injury or death. Do not proceed beyond a WARNING notice until the indicated conditions are fully understood and met.

CAUTION

A CAUTION notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in damage to the product or loss of important data. Do not proceed beyond a CAUTION notice until the indicated conditions are fully understood and met.

Earth ground symbol.

Chassis ground symbol.

WARNING

Only qualified, service-trained personnel who are aware of the hazards involved should remove the cover from the instrument.

WARNING

For continued protection against fire, replace the line fuse only with a fuse of the specified type and rating.